Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks

Fatemehsadat Mireshghallah[1], **Kartik Goyal**[2], **Archit Uniyal**[3]
Taylor Berg-Kirkpatrick[1], **Reza Shokri**[4]
[1] University of California San Diego, [2] Toyota Technological Institute at Chicago (TTIC)
[3] University of Virginia, [4] National University of Singapore
[fatemeh, tberg]@ucsd.edu,
kartikgo@ttic.edu, a.uniyal@virginia.edu, reza@comp.nus.edu.sg

Abstract

The wide adoption and application of Masked language models (MLMs) on sensitive data (from legal to medical) necessitates a thorough quantitative investigation into their privacy vulnerabilities. Prior attempts at measuring leakage of MLMs via membership inference attacks have been inconclusive, implying potential robustness of MLMs to privacy attacks. In this work, we posit that prior attempts were inconclusive because they based their attack solely on the MLM's model score. We devise a stronger membership inference attack based on likelihood ratio hypothesis testing that involves an additional reference MLM to more accurately quantify the privacy risks of memorization in MLMs. We show that masked language models are indeed susceptible to likelihood ratio membership inference attacks: Our empirical results, on models trained on medical notes, show that our attack improves the AUC of prior membership inference attacks from 0.66 to an alarmingly high 0.90 level.

1 Introduction

BERT-based encoders with Masked Language Modeling (MLM) Objectives (Devlin et al., 2018; Liu et al., 2019) have become models of choice for use as pre-trained models for various Natural Language Processing (NLP) classification tasks (Wang et al., 2018; Zhang et al., 2019; Rogers et al., 2020) and have been applied to diverse domains such as disease diagnosis, insurance analysis on financial data, sentiment analysis for improved user experience, etc (Yang et al., 2020; Gu et al., 2021; Lee et al., 2020). Given the sensitivity of the data used to train these models, it is crucial to conceive a framework to systematically evaluate the leakage of training data from these models (Shokri, 2022; Carlini et al., 2019; Murakonda and Shokri, 2020; Mireshghallah et al., 2020), and limit the leakage. The conventional way to measure the leakage of training data from machine learning models is by performing membership inference attacks (Shokri et al., 2017; Nasr et al., 2021), in which the attacker tries to determine whether a given sample was part of the training data of the target model or not. These attacks expose the extent of memorization by the model at the level of individual samples. Prior attempts at performing membership inference and reconstruction attacks on masked language models have either been inconclusive (Lehman et al., 2021), or have (wrongly) concluded that memorization of sensitive data in MLMs is very limited and these models are more private than their generative counterparts (e.g., autoregressive language models) (Vakili and Dalianis, 2021; Jagannatha et al., 2021; Nakamura et al., 2020).

We hypothesize that prior MLM attacks have been inconclusive because they rely solely on the target model's (model under attack) loss on each individual sample as a proxy for how well the model has memorized that sample. If the loss is lower than a threshold, the sample is predicted to be a member of the training set. However, the target model's loss includes confounding factors of variation like the intrinsic complexity of the sample – and thus provides a limited discriminative signal for membership prediction. This scheme has either a high false-negative rate (with a conservative threshold) – classifying many hard-to-fit samples from the training set as non-members, or a high false-positive rate (with a generous threshold) – failing to identify easy-to-fit samples that are not in the training set.

Reference-based likelihood ratio attacks, on the other hand, when applied to certain probabilistic graphical models and classifiers, have been shown to alleviate this problem and more accurately distinguish members from non-members (Murakonda et al., 2021; Ye et al., 2021). In such attacks, instead of the loss of the model under attack, we look at the ratio of the likelihood of the sample under the target model and a reference model trained on samples from the underlying population distribution that generates the training data for the target model.

This ratio recalibrates the test statistic to explain away spurious variation in model's loss for different samples due to the intrinsic complexity of the samples. Unlike most other models (e.g., generative models), however, computing the likelihood of MLMs is not straightforward. In this paper, we propose a principled framework for measuring information leakage of MLMs through likelihood ratio-based membership inference attacks and perform an extensive analysis of memorization in such models. To compute the likelihood ratio of the samples under the target and the reference MLMs, we view the MLMs as energy-based probabilistic models (Goyal et al., 2022) over the sequences. This enables us to perform powerful inference attacks on conventionally non-probabilistic models like masked language models.

We evaluate our proposed attack on a suite of masked clinical language models, following (Lehman et al., 2021). We compare our attack with the baseline from the prior work that relies solely on the loss of the target model (Yeom et al., 2018; Song and Raghunathan, 2020; Jagannatha et al., 2021). We empirically show that *our attack improves the AUC from* 0.66 *to* 0.90 on the ClinicalBERT-Base model, and achieves *a true positive rate (recall) of* 79.2% (for a false positive rate of 10%), which is a substantial improvement over the baseline with 15.6% recall. This shows that, contrary to prior results, masked language models are significantly susceptible to attacks exploiting the leakage of their training data. In low error regions (at 1% false positive rate) *our attack is* $51\times$ *more powerful than the prior work*.

We also present analyses of the effect of the size of the model, the length of the samples, and the choice of the reference model on the success of the attack. Finally, we attempt to identify features of samples that are more exposed (attack is more successful on), and observe that samples with multiple non-alphanumeric symbols (like punctuation) are more prone to being memorized. We provide instructions on how to request access to the data and code in Appendix A.2.1.

2 Membership Inference Attacks

In this section, we first formally describe the membership inference attack, how it can be conducted using likelihood ratio tests and how we apply the test for masked language models (MLMs) which do not explicitly offer an easy-to-compute probability distribution over sequences. Finally, we describe all the steps in our attack, as summarized in Figure 1.

2.1 Problem Formulation

Let M_θ denote a model with parameters θ that have been trained on data set D, sampled from the general population distribution p. Our goal is to quantify the privacy risks of releasing M_θ for the members of training set D.

We consider an adversary who has access to the target model M_θ. We assume this adversary can train a (reference) model M_{θ_R} with parameters θ_R on independently sampled data from the general population p. In a Membership Inference Attack (MIA), the objective of the adversary is to create a decision rule that determines whether a given sample s was used for training M_θ. To test the adversary, we perform the following experiment. We sample a datapoint s from either the general population or the training data with a 0.5 probability, and challenge the adversary to tell if s is selected from the training set (it is a member) or not (it is a non-member) (Murakonda et al., 2021). The precision of the membership inference attack indicates the degree of information leakage from the target model about the members of its training set. We measure the adversary's success using two metrics: (1) the adversary's power (the true positive rate), and (2) the adversary's error (the false positive rate).

2.2 Likelihood Ratio Test

Before discussing our proposed attack for MLMs in the next section, we summarize the likelihood ratio test here which forms the core of our approach. A likelihood ratio test distinguishes between a null hypothesis and an alternative hypothesis via a test statistic based on the ratio of likelihoods under the two hypotheses. Prior work demonstrated an MIA attack based on the likelihood ratio to be optimal for probabilistic graphical models (Bayesian networks) (Murakonda et al., 2021). Given a sample s from the training data of the target model, the adversary aims at distinguishing between two hypotheses:

1. Null hypothesis (H_{out}): The target sample s is drawn from the general population p, independently from the training set D.
2. Alternative hypothesis (H_{in}): The target sample s is drawn from the target model's training set D.

The goal of hypothesis testing is to find whether there is enough evidence to reject H_{out} in favor of H_{in}. We use a likelihood ratio for this purpose

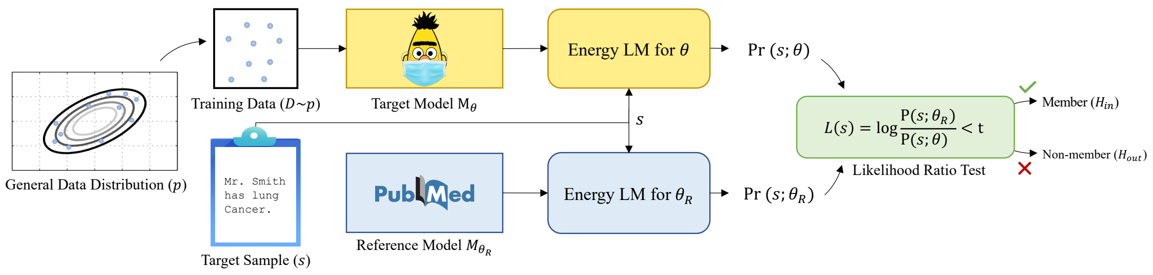


Figure 1: Overview of our attack: to determine whether a target sample s is a member of the training data ($D\sim p$) of the target model (M_θ), we feed it to the energy function formulation of M_θ so that we can compute $\Pr(s;M_\theta)$, the probability of s under M_θ. We do the same with a reference model M_{θ_R} which is trained on a disjoint data set from the same distribution as the training data. Then, we compute likelihood ratio $L(s)$, and based on this ratio and a given test threshold t, we decide if s is a member of D (H_{in}) or not (H_{out}).

which involves comparison of the likelihood of the target sample under the settings for H_{out} and H_{in} respectively. For H_{in}, we already have access to the target model, which is parameterized by θ and trained on D. For H_{out}, we require access to a model trained on the general population. As mentioned earlier, the adversary has access to a reference model parameterized by θ_R. Therefore, the likelihood ratio test is characterized by the following statistic:

$$L(s) = \log\left(\frac{p(s;\theta_R)}{p(s;\theta)}\right) \tag{1}$$

The Likelihood Ratio (LR) test is a comparison of the log-likelihood ratio statistic $L(s)$ with a threshold t. If $L(s) \le t$, then the adversary rejects H_{out} (decides in favor of membership of $s \in D$); otherwise the adversary fails to reject H_{out}. We discuss the details of selecting the threshold and quantifying the attack's success in Section 2.4.

2.3 Likelihood Ratio Test for MLMs

Performing a likelihood ratio test with masked language models is difficult because these models do not explicitly define an easy-to-compute probability distribution over natural language sequences. Following prior work (Goyal et al., 2022), we alternatively view pre-trained MLMs as energy-based probability distributions on sequences, allowing us to directly apply the likelihood ratio formalism. An energy-based sequence model defines the probability distribution over the space of possible sequences $\mathcal{S}$ as:

$$p(s;\theta) = \frac{e^{-E(s;\theta)}}{Z_\theta},$$

where $E(s;\theta)$ refers to the scalar energy of a sequence s that is parametrized by θ, and $Z_\theta = \sum_{s' \in \mathcal{S}} e^{-E(s';\theta)}$ denotes the intractable noramlization constant. Under this framework, the likelihood ratio test statistic (Eq. 1) is:

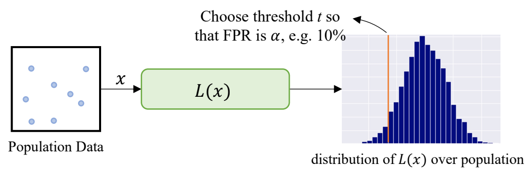
$$\begin{aligned} L(s) &= \log\left(\frac{p(s;\theta_R)}{p(s;\theta)}\right) \\ &= \log\left(\frac{e^{-E(s;\theta_R)}}{Z_{\theta_R}}\right) - \left(\log\frac{e^{-E(s;\theta)}}{Z_\theta}\right) \\ &= -E(s;\theta_R) - \log(Z_{\theta_R}) + E(s;\theta) + \log(Z_\theta) \\ &= E(s;\theta) - E(s;\theta_R) + \text{constant} \end{aligned}$$

Above, we make use of the fact that for two fixed models (i.e., target model θ, and reference model θ_R), the intractable term $\log(Z_\theta) - \log(Z_{\theta_R})$ is a global constant and can be ignored in the test. Therefore, computation of the test statistic only relies on the difference between the energy values assigned to sample s by the target model M_θ, and the reference model M_{θ_R}.

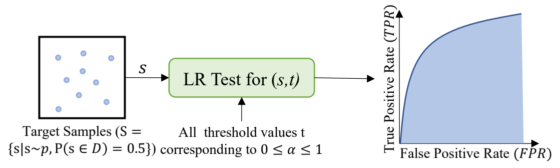
In practice, we cast a traditional MLM as an energy-based language model using a slightly different parameterization than explored by Goyal et al. (2022). Since the training of most MLMs (including the ones we attack in experiments) involves masking 15% of the tokens in a training sequence, we define our energy parameterization on these 15% chunks. Specifically, for a sequence of length T, and the subset size $l = \lceil 0.15 \times T \rceil$, we consider computing the energy with the set $\mathcal{C}$ consisting of all $\binom{T}{l}$ combinations of masking patterns.

$$E(s;\theta) = -\frac{1}{|\mathcal{C}|}\sum_{I \in \mathcal{C}}\sum_{i \in I}\log\left(p_{\text{mlm}}(s_i | s_{\setminus I};\theta)\right) \tag{2}$$

where $s_{\setminus I}$ is the sequence s with the l positions in I masked. Computing this energy, which involves running $|\mathcal{C}| = \binom{T}{l}$ forward passes of the MLM, is expensive. Hence, we further approximate this parametrization by summing up over K random masking patterns where $K \ll |\mathcal{C}|$.


Choose threshold t so that FPR is α, e.g. 10%
x
$L(x)$
Population Data
distribution of $L(x)$ over population

(a) Selecting threshold t


s
LR Test for (s,t)
Target Samples (S = {s|s~p, P(s ∈ D) = 0.5})
All threshold values t corresponding to $0 \leq \alpha \leq 1$
True Positive Rate (TPR)
False Positive Rate (FPR)

(b) Plotting the ROC curve

Figure 2: (a) Selecting a threshold for the attack using population data and (b) plotting the ROC curve to show the true-positive vs. false-positive rate trade-off, given different thresholds.

2.4 Quantifying the Privacy Risk

Given the form of the likelihood ratio test statistic (Eq. 2) and energy function formulation for MLM likelihood (Eq. 2), we conduct the attack as follows (shown in Figure 1):

1. Given a sample s whose membership we want to determine, we calculate its energy $E(s; \theta)$ under the model under attack (M_θ) using Eq. 2. We calculate the energy $E(s; \theta_R)$ under the reference model. Using Eq. 1, we compute the test statistic $L(s)$ by subtracting the two energies.

2. We compare $L(s)$ to a threshold t, and if $L(s) \leq t$, we reject the null hypothesis (H_{out}) and mark the sample as a member. Otherwise, we mark it as a non-member.

Choosing the threshold. The threshold determines the (false positive) error the adversary is willing to tolerate in the membership inference attack. Thus, for determining the threshold t, we select a false positive rate α, and empirically compute t as the corresponding percentile of the likelihood ratio statistic over random samples from the underlying distribution. This process is visualized in Figure 2a. We empirically estimate the distribution of the test statistic $L(x)$ using all the sequences x drawn from the general population distribution. This yields the distribution of L under the null hypothesis. We then select the threshold such that the tolerance of attack's error i.e. the rate at which attack *falsely* classifies the population data as "members" is $\alpha\%$.

Quantifying the Privacy Risk. The attacker's success (i.e. the privacy loss of the model) can be quantified using the relation between the attack's power (the true positive rate) versus its error (the false positive rate). Higher power for lower errors indicates larger privacy loss. To compare two attack algorithms (e.g., our method versus the target model loss based methods), we can compute their power for all different error values, which can be illustrated in an ROC curve (as in Figure 2b and Figure 4). This enables a complete comparison between two attack algorithms. The Area Under the Curve (AUC) metric for each attack provides an overall threshold independent evaluation of the privacy loss under each attack.

3 Experimental Setup

We conduct our experiments using the pre-processed data, and pre-trained models provided by Lehman et al. (2021). We use this medical-based setup as medical notes are sensitive and leakage of models trained on notes can cause privacy breaches. In this section, we briefly explain the details of our experimental setup. Appendix A.2 provides more details. Table 1 provides a summary.

3.1 Datasets

We run our attack on two sets of target samples, in both of which the "members" portion is sampled from the training set (D) of our target models, which is the MIMIC-III dataset. The non-members, however, are different. For the results shown under "MIMIC", the non-members are a held-out subset of the MIMIC data that was not used in training. For i2b2, the non-members are from a different (but similar) dataset, i2b2. Below we elaborate on each of these datasets (full detail in Appendix A.2.2). Both the datasets require a license for access, so we cannot show examples of the training data.

MIMIC-III. The target models we attack are trained on the pseudo re-identified MIMIC-III notes which consist of 1,247,291 electronic health records (EHR) of 46,520 patients.

i2b2. This dataset was curated for the i2b2 de-identification of protected health information (PHI) challenge in 2014 (Stubbs and Özlem Uzuner, 2015). We use this dataset as a secondary non-member dataset since it is similar in domain to MIMIC-III (both are medical notes), is larger in terms of size than the held-out MIMIC-III set, and has not been used as training data for our models.

3.2 Models

Target Models. We perform our attack on 4 different pre-trained ClinicalBERT models, that are all trained on MIMIC-III, but with different training

Table 1: Summary of model and baseline notations used in the results.

	Notation	Explanation
Models	Base	ClinicalBERT-base target model, trained for 300k iterations w/ sequence length 128 and 100k iterations w/ sequence length 512.
	Base++	ClinicalBERT++ target model, same as the Base model but trained for longer: trained for 1M iterations w/ a sequence length of 128.
	Large	ClinicalBERT-large target model, trained for 300k iterations w/ sequence length 128 and 100k iterations w/ sequence length 512.
	Large++	ClinicalBERT-large++ target model, same as the Large model but trained for longer: trained for 1M iterations w/ a sequence length of 128.
Methods	(A) w/ μ thresh.	Baseline with threshold set to be the mean of training sample losses (μ) (for reporting threshold-dependant metrics)
	(A) w/ Pop. thresh.	Baseline with threshold set so that there is 10% false positive rate (for reporting threshold-dependant metrics)
	(B) w/ Pop. thresh.	Our method with threshold set so that there is 10% false positive rate (for reporting threshold-dependant metrics)

procedures, summarized in Table 1 under Models.

Reference Models. We use `Pubmed-BERT` [1] trained on pre-processed PubMed texts containing around 4000M words extracted from PubMed ASCII code version (Peng et al., 2019) as our main domain-specific reference model, since its training data is similar to MIMIC-III in terms of domain, however, it does not include MIMIC-III training data. We also use the standard pre-trained `bert-base-uncased` as a general-domain reference model for ablating our attack.

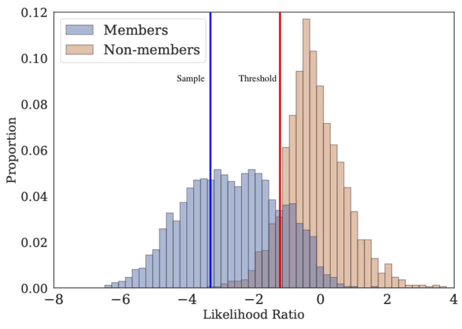
3.3 Baselines

We compare our results with a popular prior method, which uses the loss of the target model as a signal to predict membership (Yeom et al., 2018; Jayaraman et al., 2021; Ye et al., 2021). We show this baseline as *Model loss* in our tables. This baseline could have two variations, based on the way its threshold is chosen: (1) μ *threshold* (Jagannatha et al., 2021), which assumes access to the mean of the training data loss, μ and uses it as the threshold for the attack, and (2) *population threshold (pop. thresh.)* which calculates the loss on a population set of samples (samples that were not used in training but are similar to training data), and then selects the threshold that would result in a 10% false positive rate on that population.

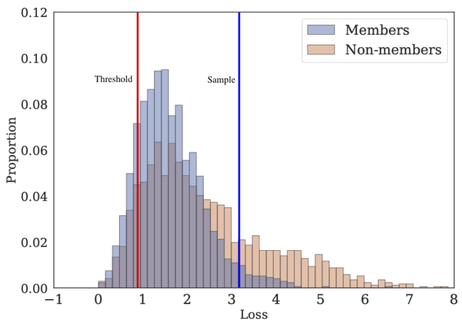
3.4 Metrics

Area Under the ROC Curve (AUC). The ROC curve is a plot of power (true positive rate) versus error (false positive rate), measured across different thresholds t, which captures the trade-off between power and error. Thus, the area under the ROC curve (AUC) is a single, threshold-independent metric for measuring the strength of the attack. Figure 2b shows how we obtain the ROC curve. AUC $= 1$ implies that the attacker can correctly classify all target samples as members or non-members.

Precision and Recall. We set $\alpha = 10\%$ as the false positive rate and choose the threshold accordingly, as shown in Fig. 2a. For precision, we measure the percentage of samples correctly inferred as members of the training set out of the total number of target samples inferred as members by the attack. For recall, we measure the percentage of samples correctly inferred as members of the training set out of the total number of target samples that are actually members of the training set.



(a) Likelihood Ratio $L(s)$ Histogram



(b) Target Model Loss Histogram

Figure 3: (a) likelihood ratio histogram for training data members and non-members. (b) loss histogram for training data members and non-members. The blue lines in the two figures correspond to the same target sample (which is a random member of training-set). The red line is is the threshold at $\alpha = 10\%$ false positive rate. The threshold from our attack (a) is correctly able to label the test sample as a training-set member but the thresholds from the baseline attack (b) fails to do so.

4 Results

In this section, we discuss our experimental results and main observations. First, we explore the overall performance improvement of our approach over baselines. Later, we analyze the effectiveness of our approach across several factors of variation that have an effect on the leakage of the model. (e.g. length of samples, model size, including names

[1] `bionlp/bluebert_pubmed_uncased_L-12_H-768_A-12`

Table 2: Overview of our attack on the ClinicalBERT-Base model, using PubMed-BERT as the reference. Sample-level attack attempts to determine membership of a single sample, whereas patient-level determines membership of a patient based on all their notes. The MIMIC and i2b2 columns determine which dataset was used as non-members in the target sample pool.

		Sample-level		Patient-level	
	Non-members	MIMIC	i2b2	MIMIC	i2b2
AUC.	(A) Model loss	0.662	0.812	0.915	1.000
	(B) Ours	0.900	0.881	0.992	1.000
Prec.	(A) w/ μ thresh.	61.5	77.6	87.5	100.0
	(A) w/ Pop. thresh.	61.2	79.6	87.5	92.5
	(B) w/ Pop. thresh.	88.9	87.5	93.4	92.5
Rec.	(A) w/ μ thresh	55.7	55.8	49.5	49.5
	(A) w/ Pop. thresh.	15.6	39.0	49.5	100.0
	(B) w/ Pop. thresh.	79.2	69.9	100.0	100.0

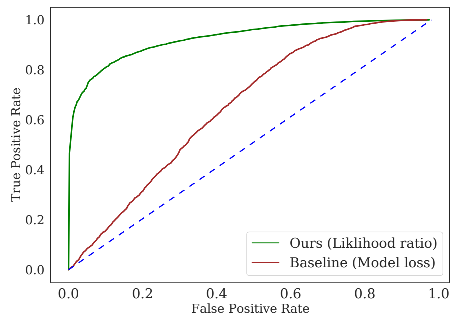


Figure 4: The ROC curve of sample-level attack on Clinical-BERT with MIMIC used as non-member. Green line shows our attack and the red line shows the baseline loss-based attack. The blue dashed line shows AUC=0.5 (random guess). This figure corresponds to the results presented in the first column of Table 2.

etc.) Finally, we explore correlations between samples that are deemed to be *exposed* by our approach. We provide further studies and ablations on choosing a lower false-positive rate of 1%, using different energy formulation and changing target sequence lengths in Appendix sections A.3.1, A.3.2, and A.3.3, respectively.

4.1 Comparison with Baseline

Table 2 shows the metrics for our attack and the baseline's on both sample and patient level, with held-out MIMIC-III and i2b2 medical notes used as non-member samples (Figure 4 shows the ROC curve). The table shows that our method significantly outperforms the target model loss-based baselines (Jagannatha et al., 2021; Yeom et al., 2018), which threshold the loss of the target model based on either the mean of the training samples' loss (μ), or the population samples' loss. Our attack's improvement over the baselines is more apparent in the case where both the members and non-members are from MIMIC-III. This case is harder for the baselines since members and non-members are much more similar and harder to distinguish if we only look at the loss of the target model. Our attack, however, is successful due to the use of a reference, which helps magnify the gap in the behavior of the target model towards members and non-members, thereby teasing apart similar samples.

We can also see that in terms of precision/recall trade-off, our attack has a consistently higher recall, with an average higher precision. Population loss based thresholding ((A) w/ Pop. thresh.) has the lowest recall of 15.6%, which is due to members and non-members achieving similar losses from the target model due to their similarity. This is also shown in Figure 3b. In Figure 3a, however, we see a distinct separation between the member and non-member histogram distributions when we use the ratio statistic $L(s)$ as the test criterion for our attack. This results in the estimation of a useful threshold that correctly classifies the blue line sample as a member, as opposed to using only the target model loss (Figure 3b). Finally, we observe that all the metrics have higher values on the patient-level attack, compared to sample-level, for both our attack and the baselines. This is due to the higher granularity of the patient level attack, as it makes the decision based on *an aggregate* of multiple samples.

Table 3: Effect of target sample length: Sample-level attack on the ClinicalBERT-Base model, using PubMed-BERT as the reference. The MIMIC and i2b2 columns determine which dataset was used as non-members in the target sample pool. Short and long show a break down of the length of target samples.

		Short		Long	
	Non-members	MIMIC	i2b2	MIMIC	i2b2
AUC.	(A) Model loss	0.516	0.756	0.662	0.812
	(B) Ours	0.830	0.845	0.900	0.881
Prec.	(A) w/ μ thresh.	50.9	70.0	61.5	77.6
	(A) w/ Pop. thresh.	42.0	72.5	61.2	79.6
	(B) w/ Pop. thresh.	87.3	86.3	88.9	87.5
Rec.	(A) w/ μ thresh.	55.3	55.3	55.7	55.8
	(A) w/ Pop. thresh.	7.2	26.3	15.6	39.0
	(B) w/ Pop. thresh.	68.2	62.9	79.2	69.9

4.2 Effect of Sample Length and Model Size

Tables 3 and 4 show the metrics for our attack and the baseline broken down based on the length of the target sample, and the size and training epochs of the target model, respectively. In Table 3, the target model is same as that of Table 2, ClinicalBERT-base. Short samples are those that have between 10 to 20 tokens, and long samples have 20 to 60 tokens.

Table 4: Effect of model size and training: Sample-level attack on the four different ClinicalBERT models, using PubMed-BERT as the reference and the MIMIC data as non-members. Base++ (Large++) is same as Base (Large), but trained for more epochs.

	Target Model	Base	Base++	Large	Large++
AUC.	(A) Model loss	0.662	0.656	0.679	0.700
	(B) Ours	0.900	0.894	0.904	0.905
Prec.	(A) w/ μ thresh.	61.5	61.0	62.4	64.9
	(A) w/ Pop. thresh.	61.2	61.2	63.9	69.1
	(B) w/ Pop. thresh.	88.9	88.8	88.9	88.9
Rec.	(A) w/ μ thresh.	55.7	55.8	56.4	56.1
	(A) w/ Pop. thresh.	15.6	15.6	17.6	22.2
	(B) w/ Pop. thresh.	79.2	78.5	79.2	79.3

We can see that both the baseline and our attacks show more leakage for long sentences than they do for short sequences, which could be due to the longer sentences being more unique and thus being more likely to provide a discriminative signal for a sequence-level decision. Table 4 shows the attacks mounted on the four models from Table 1. We see that leakage on all the models is very similar, however, the AUC on Large++ is consistently higher than on Base, which hints at the observation made by (Carlini et al., 2021b) that larger models tend to have a higher capacity for memorization.

4.3 Effect of Changing the Reference Model

Table 5 studies how changing the reference model would affect the success of the attack. Here, *Pubmed* is the reference model that is used in the previous experiments, and *BERT-base* is Huggingface's pre-trained BERT. We observe that the attack using BERT-base performs well, but is worse than using Pubmed, especially in terms of recall (true positive rate). The main reason behind this is the domain overlap between the Pubmed reference model and the model under attack. An ideal reference model for this attack would be trained on data from a domain that is similar to that of the target model's training set so as to better characterize the intrinsic complexity of the samples. On the other hand, a reference model trained on a different data distribution (in this case Wikipedia) would give the same score to easy and difficult samples, thereby decreasing the true positive rate (recall), as shown in the table.

4.3.1 Effect of Inserting Names

Table 6 shows results for attacking the name insertion model (Lehman et al., 2021), shown as Base-b, where the patient's first and last name are prepended to each training sample. We see that our attack's performance is better on the name-insertion model,

Table 5: Effect of reference model: Sample-level attacks on ClinicalBERT-Base model, using PubMed-BERT and standard bert-base-uncased as the reference and MIMIC data as non-member.

		Base		Large++	
	Reference Model	pubmed	bert	pubmed	bert
AUC.	(A) Model loss	0.662	0.662	0.700	0.700
	(B) Ours	0.900	0.883	0.905	0.889
Prec.	(A) w/ μ thresh.	61.5	61.5	64.9	64.9
	(A) w/ Pop. thresh.	61.2	61.2	69.1	69.1
	(B) w/ Pop. thresh.	88.9	87.8	88.9	88.0
Rec.	(A) w/ μ thresh.	55.7	55.7	56.1	56.1
	(A) w/ Pop. thresh.	15.6	15.6	22.2	22.2
	(B) w/ Pop. thresh.	79.2	71.5	79.3	72.6

Table 6: Effect of inserting names: Sample and Patient-level attacks on ClinicalBERT-Base and Base-b (name insertion) model, using PubMed-BERT as the reference and MIMIC data as non-member. We study the effect that inserting names into all training samples has on the leakage of the model.

		Sample-level		Patient-level	
	Target model	Base	Base-b	Base	Base-b
AUC.	(A) Model loss	0.662	0.561	0.915	0.953
	(B) Ours	0.900	0.960	0.992	1.000
Prec.	(A) w/ μ thresh.	61.5	53.0	87.5	100.0
	(A) w/ Pop. thresh.	61.2	44.1	87.5	91.1
	(B) w/ Pop. thresh.	88.9	90.2	93.4	92.5
Rec.	(A) w/ μ thresh.	55.7	54.0	49.5	48.5
	(A) w/ Pop. thresh.	15.6	7.8	49.5	82.8
	(B) w/ Pop. thresh.	79.2	91.3	100.0	100.0

compared to the base model, whereas the baseline attack performs worse (in the sample-level scenario). We hypothesize that this is due to the "difficulty" of the samples. Adding names to the beginning of each sample actually increases the entropy of the dataset overall, since in most cases they don't have a direct relation with the rest of the sentence (except for very few sentences that directly state a person's disease), therefore they might as well be random. This makes these sentences more difficult and harder to learn, as there is no easy pattern. Hence, on average, these sentences have higher loss values (2.14 for name inserted samples, vs. 1.61 for regular samples). However, for the non-members, since they don't have names attached to them, the average loss is the same (the 10% FPR threshold is 1.32), and that is why the attack performs poorly on these samples, as most of the members get classified as non-members. For our attack, since we use the reference, we are able to tease apart such hard samples as they are extremely less likely given the reference than they are given the target model.

4.4 Correlations between Memorized Samples

To evaluate whether there are correlations between samples that have high leakage based on our attack (i.e. training samples that are successfully detected as members), we conduct an experiment. In this experiment, we create a new train and test dataset, by subsampling the main dataset and selecting 5505 and 7461 samples, respectively. We label the training and test samples based on whether they are exposed or not, i.e. whether the attack successfully detects them as training samples or not, and get 2519 and 3283 samples labeled as “memorized”, for the train and test set. Since our goal is to see if we can find correlations between the memorized samples of the training set and use those to predict memorization on our test set, we create features for each sample, and then use those features with the labels to create a simple logistic regression classifier that predicts memorization.

Table 7 shows these results in terms of precision and recall for predicting if a sample is “memorized" or not, with different sets of features. The first 4 rows correspond to individual handcrafted feature sets: (A) the number of digits in the sample, (B) length of a sample (in tokens), (C) the number of non-alphanumeric characters (this would be characters like ’*’, ’-’, etc.). (D) corresponds to feature sets that are obtained by encoding the tokenized sample by the frequency of each of its tokens, and then taking the 3 least frequent tokens’ frequencies as features (the frequency comes from a frequency dictionary built on the training set). We can see that among the hand-crafted features, (C) is most indicative, as it counts the characters that are more out-of-distribution and are possibly not determined by grammatical rules or consistent patterns. (C) and (D) concatenated together perform slightly better than (C) alone, which could hint at the effect frequency of tokens and how common they are could have on memorization. We also get a small improvement over these by concatenating (B), (C), and (D), which shows the length has a slight correlation too.

Table 7: Analysis of correlations between samples that are leaked through our attack. We want to see what features are shared among all leaked samples by extracting a list of possible features and training a simple logistic regression model on a subset of the original training data (D), and then testing it on another subset. The logistic regression model tries to predict whether a sample would be leaked or not (based on whether our model has classified it as a member or not). The precision and recall here are those of the logistic regression model, for predicting leaked training samples.

Features	Train		Test	
	Prec.	Rec.	Prec.	Rec.
(A) #Digits	0.0	0.0	0.0	0.0
(B) Seq. Len	0.0	0.0	0.0	0.0
(C) #Non-alphanumeric	71.2	46.6	69.2	47.5
(D) 3 Least Frequent	68.9	40.5	63.8	39.2
(C) & (D)	73.9	58.8	71.1	57.8
(B) & (C) & (D)	74.3	61.3	72.1	61.3
(A) & (B) & (C) & (D)	74.3	61.3	72.1	61.3

5 Related Work

Prior work on measuring memorization and leakage in machine learning models can be classified into two main categories: (1) membership inference attacks and (2) training data extraction attacks.

Membership inference. Membership Inference Attacks (MIA) try to determine whether or not a target sample was used in training a target model (Shokri et al., 2017; Yeom et al., 2018). These attacks can be seen as privacy risk analysis tools (Murakonda and Shokri, 2020; Nasr et al., 2021; Kandpal et al., 2022), which help reveal how much the model has memorized the individual samples in its training set, and what the risk of individual users is (Nasr et al., 2019; Long et al., 2017; Salem et al., 2018; Ye et al., 2021; Carlini et al., 2021a). A group of these attacks rely on behavior of shadow models to determine the membership of given samples (Jayaraman et al., 2021; Shokri et al., 2017). Song and Shmatikov mounts such an attack on LSTM-based text-generation models, Mahloujifar et al. mounts one on word embedding, Hisamoto et al. applies it to machine translation and more recently, Shejwalkar et al. mounts it on transformer-based NLP classification models. Mounting such attacks is usually costly, as their success relies upon training multiple shadow models on different partitionings of shadow data, and access to adequate shadow data for training such models.

Another group of MIAs relies solely on the loss value of the target sample, under the target model, and thresholds this loss to determine membership (Jagannatha et al., 2021; Yeom et al., 2018). Song and Raghunathan mount such an attack on word embedding, where they try to infer if given samples were used in training different embedding models. Jagannatha et al. (2021), which is the work closest to ours, uses a thresholding loss-based attack to infer membership on MLMs. Our approach instead incorporates a reference model by using an energy-based formulation to mount a likelihood ratio based attack and achieves higher AUC as shown in the results.

Training data extraction. Training data extraction quantifies the risk of extracting training data by probing a trained language model (Salem et al., 2020; Carlini et al., 2019; Zanella-Béguelin et al., 2020; Carlini et al., 2021b, 2022; Nakamura et al., 2020). One such prominent attacks on NLP models is that of Carlini et al. (2021b), where they take more than half a million samples from different GPT-2 models, sift through the samples using a membership inference method to find samples that are most likely to have been memorized. Lehman et al. (2021) mount the same data extraction attack on MLMs, but their results are inconclusive as to how much MLMs memorize samples. They also mount other types of attacks, where they try to extract a person's name given their disease, or disease given name, but in all their attacks, they only use signals from the target model and consistently find that a frequency-based baseline (i.e. one that would always guess the most frequent name/disease) is more successful.

6 Conclusions

In this paper, we introduce a principled membership inference attack based on likelihood ratio testing to measure the training data leakage of Masked Language Models (MLMs). In contrast to prior work on MLMs, we rely on signals from both the model under attack and a reference model to decide the membership of a sample. This enables performing successful membership inference attacks on data points that are hard to fit, and therefore cannot be detected using the prior work. We also perform an analysis of *why* these models leak, and which data points are more susceptible to memorization. Our attack shows that MLMs are significantly prone to memorization. This work calls for designing robust privacy mitigation algorithms for such language models.

Limitations

Membership inference attacks form the foundation of privacy auditing and memorization analysis in machine learning. As we show in this paper, and as it is shown in the recent work (Carlini et al., 2021a; Ye et al., 2021), these attacks are very efficient in identifying privacy vulnerabilities of models with respect to individual data records. However, for a thorough analysis of data privacy, it is not enough to rely only on membership inference attacks. We thus would need to extend our analysis to reconstruction attacks and property inference attacks.

Ethics Statement

We use two datasets in this paper, MIMIC-III and i2b2, both of which contain sensitive data and can only be accessed by request[2] and after agreeing to the data usage and confidentiality terms[3] and passing proper training for ethical and privacy-preserving use of the data. For reproduction of our results, code will be made available only by request and for research purposes, only to researchers who provide proof of authorized access to the datasets (by forwarding the access granted emails from MIMIC-III and i2b2 to the first author[4]).

To protect models against membership inference attacks, like the one proposed in this work, differentially private training algorithms (Abadi et al., 2016; Chaudhuri et al., 2011) can be used, as they are theoretically designed to protect the membership of each data record individually. Other methods such as adversarial training (Mireshghallah et al., 2021) and personally identifiable information scrubbing (Dernoncourt et al., 2017) can also be used, however, they do not provide the worst-case guarantees that differential privacy does (Brown et al., 2022).

Acknowledgements

The authors would like to thank the anonymous reviewers and meta-reviewers for their helpful feedback. We also thank our colleagues at the UCSD Berg Lab and NUS for their helpful comments and feedback.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr.

[2]Access can be requested through https://mimic.mit.edu/docs/gettingstarted/ and https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

[3]Data Usage Agreements (DUA) are available in https://physionet.org/content/mimiciii/view-dua/1.4/ and https://projects.iq.harvard.edu/files/n2c2/files/n2c2_data_sets_dua_preview_-_academic_user.pdf for the datasets, respectively.

[4]fmireshg@eng.ucsd.edu

2022. What does it mean for a language model to preserve privacy? *arXiv preprint arXiv:2202.05520*.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2021a. Membership inference attacks from first principles.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021b. Extracting training data from large language models.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3).

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2022. Exposing the implicit energy networks behind masked language models via metropolis–hastings. In *International Conference on Learning Representations*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership Inference Attacks on Sequence-to-Sequence Models: Is My Data In Your Machine Translation System? *Transactions of the Association for Computational Linguistics*, 8:49–63.

Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2021. Membership inference attack susceptibility of clinical language models.

Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. 2021. Revisiting membership inference under realistic assumptions.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. *arXiv preprint arXiv:2202.06539*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online. Association for Computational Linguistics.

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

Yunhui Long, Vincent Bindschaedler, and Carl A. Gunter. 2017. Towards measuring membership privacy. *ArXiv*, abs/1712.09136.

Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. 2021. Membership inference on word embedding and beyond. *arXiv preprint arXiv:2106.11384*.

Fatemehsadat Mireshghallah, Huseyin Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. 2021. Privacy regularization: Joint privacy-utility optimization in languagemodels. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3799–3807.

Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. 2020. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254*.

Sasi Kumar Murakonda and Reza Shokri. 2020. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*.

Sasi Kumar Murakonda, Reza Shokri, and George Theodorakopoulos. 2021. Quantifying the privacy risks of learning high-dimensional graphical models. In *International Conference on Artificial Intelligence and Statistics*, pages 2287–2295. PMLR.

Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2020. Kart: Privacy leakage framework of language models pre-trained with clinical records.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE.

Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 866–882. IEEE.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-Leak: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1291–1308. USENIX Association.

Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *ArXiv*, abs/1806.01246.

Virat Shejwalkar, Huseyin A Inan, Amir Houmansadr, and Robert Sim. 2021. Membership inference attacks against nlp classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*.

Reza Shokri. 2022. Auditing data privacy for machine learning. Santa Clara, CA. USENIX Association.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

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

Congzheng Song and Vitaly Shmatikov. 2018. The natural auditor: How to tell if someone used your words to train their model. *ArXiv*, abs/1811.00513.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Thomas Vakili and Hercules Dalianis. 2021. Are clinical bert models privacy preserving? the difficulty of extracting patient-condition associations. In *Proceedings of the AAAI 2021 Fall Symposium on Human Partnership with Medical AI : Design, Operationalization, and Ethics (AAAI-HUMAN 2021)*, number 3068 in CEUR Workshop Proceedings.

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

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2021. Enhanced membership inference attacks against machine learning models. *arXiv preprint arXiv:2111.09679*.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. *Analyzing Information Leakage of Updates to Natural Language Models*, pages 363–375. Association for Computing Machinery, New York, NY, USA.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Table 8: Summary of our attack's notations.

Notation	Explanation
D	Training dataset
S	Set of target samples whose membership we want to determine.
s	A given target sample ($s \in S, p(s \in D) = 0.5$))
R	Reference model
t	Attack threshold
L(s)	Likelihood ratio for given sample s

A Appendix

A.1 Notations

To summarize and clarify the notations used in the paper for explaining our attack, we added Table 8.

A.2 Detailed Experimental Setup

A.2.1 Code and Data Access

We use two datasets in this paper, MIMIC-III and i2b2, both of which contain sensitive data and can only be accessed by request through https://mimic.mit.edu/docs/gettingstarted/ and https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/, and after agreeing to the data usage and confidentiality terms[5] and passing proper training for ethical and privacy-preserving use of the data. For reproduction of our results, code will be made available only by request and for research purposes, only to researchers who provide proof of authorized access to the datasets (by forwarding the access granted emails from MIMIC-III and i2b2 to the first author[6]).

A.2.2 Datasets

We run our attack on two sets of target samples, one we denote by "MIMIC" and the other by "i2b2" in the results (both medical notes). For both of these, the "members" portion of the target samples is from the training set (D) of our target models, which is the MIMIC-III dataset. However, the non-members are different. For the results shown under MIMIC, the non-members are a held-out subset of the MIMIC data that was not used in training. For i2b2, the non-members are from a different (but similar) dataset, i2b2. Below we elaborate on this setup and each of these datasets.

MIMIC-III The target models we attack are trained (by Lehman et al.) on the pseudo re-identified MIMIC III notes which consist of 1, 247, 291 electronic health records (EHR) of 46,520 patients. These records have been altered such that the original first and last names are replaced with new first and last names sampled from the US Census data. Only 27,906 of these patients had their names explicitly mentioned in the EHR.

For the attack's target data, we use a held-out subset of MIMIC-III consisting of 89 patients (4072 sample sequences) whose data was not used during training of the ClinicalBERT target models and use them as "non-member" samples. For "member" samples, we take a 99 patient subsample of the entire training data, and then subsample 4072 sample sequences from that (we do this 10 times for each attack and average the results over), so that the number of member and non-member samples are the same and the target pool is balanced.

i2b2 This dataset was curated for the i2b2 de-identification of protected health information (PHI) challenge in 2014 (Stubbs and Özlem Uzuner, 2015). We use this dataset as a secondary non-member dataset, since it is similar in domain to MIMIC-III (both are medical notes), is larger in terms of size than the held-out MIMIC-III set, and has not been used as training data for our models. We subsample 99 patients from i2b2, consisting of 18561 sequences, and use them as non-members.

The population data that we use to evaluate the distribution of likelihood ratio over the null hypothesis (which is used to compute the threshold) is disjoint with the non-member set that we use to evaluate the attack. We randomly select 99 patients from the i2b2 dataset for this purpose.

A.2.3 Target Models

We perform our attack on 5 different pre-trained models, that are all trained on MIMIC-III, but with different training procedures:

ClinicalBERT (Base) BERT-base architecture trained over the pseudo re-identified MIMIC-III notes for 300k iterations for sequence length 128 and for 100k iterations for sequence length 512.

ClinicalBERT++ (Base++) BERT-base architecture trained over the pseudo re-identified MIMIC III notes for 1M iterations at a sequence length of 128.

ClinicalBERT-Large (Large). BERT-large architecture trained over the pseudo re-identified MIMIC-III notes for 300k iterations for sequence length 128 and for 100k iterations for sequence length 512.

[5]Data Usage Agreements (DUA) are available in https://physionet.org/content/mimiciii/view-dua/1.4/ and https://projects.iq.harvard.edu/files/n2c2/files/n2c2_data_sets_dua_preview_-_academic_user.pdf for the datasets, respectively.

[6]fmireshg@eng.ucsd.edu

ClinicalBERT-large++ (Large++) BERT-large architecture trained over the pseudo re-identified MIMIC III notes for 1M iterations at a sequence length of 128.

ClinicalBERT-b ((Base-b), Name Insertion). Same training and architecture as ClincalBERT-base, but that the patient's surrogate name is prepended to the beginning of every sentence. This model is used to identify the effect of name-insertion on the memorization of BERT-based models (Lehman et al., 2021).

A.2.4 Computational Resources

For this paper, we did not train any models, so the GPU training time is 0 hours. However, for getting the likelihoods, we ran inference on the sequences in our target samples pool. For that, we used an RTX2080 GPU with 11GB of memory for 18 hours.

A.3 Further Studies

A.3.1 Lower False Positive Rate

All the threshold-dependant results (precision and recalls) in Section 4 are reported with a threshold set for having $\alpha = 10\%$ false positive rate (using the mechanism shown in Figure 2a). In this section, we want to look at lower false positive rates, like we do in Figure 5, and see how well our attack does when precision is very important to us and we do not want to get any false positives. These results are shown in Table 10. (This table corresponds with Table 4 from Section 4.2 and the sample-level part of Table 6.) We can see that compared to Table 10, as we are decreasing the false positive rate, the performance gap between our attack and the baseline increases drastically, showin that our attack performs really well under tight false positive rate constraints. Note that AUC is not threshold dependant therefore it has the same value in both tables.

A.3.2 Using Normalized Energy

In the paper, we use $E(s;\theta)$, as shown in Equation 2 for finding the likelihood ratio. In other words, for finding the likelihood ratio, we basically calculate the loss of the target model and reference model (using 15% masking and averaging over 10 times) on the given sequence s, and subtract them. However, another way to approach this problem of calculating likelihood ratio is to use the normalized energy (instead of the loss) as introduced in (Goyal et al., 2022), instead of the loss. For calculating the normalized energy, we mask each token in the sequence, one token at a time (instead of 15%), calculated the loss, and average over all the tokens. To see how this energy does, we have used it to mount our attack, and we show the results in Table 11.

Compared to using loss (Table 9) it seems like the AUC for normalized energy is higher overall,

A.3.3 Results for Short Sequences

All the results in Section 4 are reported for long sequences (more than 20 tokens long), except those reported in Table 3, where we ablate the privacy risks for sequences of different lengths. In this section, for the sake of completion, we are reporting results for short sequences as well as long sequences, for all the five models we study. These results are shown in Table 9. (This table corresponds with Table 4 from Section 4.2 and the sample-level part of Table 6.)

A.3.4 Qualitative Comparison with Baseline

Figures 3a and 3b show histogram visualizations of $L(s)$ (likelihood ratio statistic) and model loss, over the target sample pool (i.e. mixture of members and non-members from the MIMIC dataset), respectively. Here the target model is ClinicalBERT-Base. The blue line represents a target query sample drawn from the member set. The point of these visualizations is to show a case in which a sample is misclassified as a non-member by the model loss baseline, but is correctly classified as a member using our attack.

The member and non-member distributions' histograms are shown via light blue and light orange bars respective. The red line indicates the threshold that is selected such that $\alpha = 0.1$, i.e. 10% false positive rate. In Figure 3a we see a distinct separation between the member and non-member histogram distributions when we use $L(s)$ as the test criterion for our attack. This results in the estimation of a useful threshold that correctly classifies the blue line sample as a member. In contrast, the baseline attack by (Jagannatha et al., 2021), in Figure 3b based solely upon the target model's loss leads to a high overlap between the member and non-member histograms which leads to a threshold that misclassifies the sample represented by the blue line. These histograms show that the reference model used in our method helps in getting a sense of how hard each sample is in general, and puts each point in perspective.

A.3.5 ROC Curve Magnified

In Figure 5 We have plotted Figure 4 from the results, but with logarithmic x-axis, to zoom in on the low false positive rate section and really show the differences between our attack and the baseline.

Table 9: Sample-level attack results (with $\alpha = 10\%$ false positive rate used for thresholding) on the four ClinicalBERT models, plus the Base-b model (name insertion). We use PubMed-BERT as the reference model and the MIMIC data as non-members. Base++ (Large++) is same as Base (Large), but trained for more epochs. Base-b is the same as the Base model, but the training data was modified to prepend patient's first and last name to each sequence. This table studies effect of model size, training and sequence length on leakage.

		Short					Long				
	Non-members	Base	Base++	Large	Large++	Base-b	Base	Base++	Large	Large++	Base-b
AUC.	(A) Model loss	0.662	0.656	0.679	0.700	0.561	0.516	0.513	0.509	0.536	0.391
	(B) Ours	0.900	0.894	0.904	0.905	0.960	0.830	0.827	0.835	0.843	0.912
Prec.	(A) w/ μ thresh.	61.5	61.0	62.4	64.9	53.0	50.9	50.6	50.7	52.5	44.3
	(A) w/ Pop. thresh.	61.2	61.2	63.9	69.1	44.1	42.0	40.6	40.0	43.4	25.9
	(B) w/ Pop. thresh.	88.9	88.8	88.9	88.9	90.2	87.3	87.3	87.3	87.3	89.3
Rec.	(A) w/ μ thresh.	55.7	55.8	56.4	56.1	54.0	55.3	55.1	56.2	55.8	54.3
	(A) w/ Pop. thresh.	15.6	15.6	17.6	22.2	7.8	7.2	6.8	6.6	7.6	3.5
	(B) w/ Pop. thresh.	79.2	78.5	79.2	79.3	91.3	68.2	68.2	68.4	68.4	82.7

Table 10: Sample-level attack results (with $\alpha = 1\%$ false positive rate used for thresholding) on the four ClinicalBERT models, plus the Base-b model (name insertion). We use PubMed-BERT as the reference model and the MIMIC data as non-members. Base++ (Large++) is same as Base (Large), but trained for more epochs. Base-b is the same as the Base model, but the training data was modified to prepend patient's first and last name to each sequence. This table studies effect of model size, training and sequence length on leakage.

		Short					Long				
	Non-members	Base	Base++	Large	Large++	Base-b	Base	Base++	Large	Large++	Base-b
AUC.	(A) Model loss	0.662	0.656	0.679	0.700	0.561	0.516	0.513	0.509	0.536	0.391
	(B) Ours	0.900	0.894	0.904	0.905	0.960	0.830	0.827	0.835	0.843	0.912
Prec.	(A) w/ μ thresh.	61.5	61.0	62.4	64.9	53.0	50.9	50.6	50.7	52.5	44.3
	(A) w/ Pop. thresh.	53.7	53.8	62.1	71.4	35.1	9.4	5.8	9.3	18.3	7.5
	(B) w/ Pop. thresh.	98.5	98.4	98.4	98.4	98.8	97.9	97.9	98.0	97.9	98.4
Rec.	(A) w/ μ thresh.	55.7	55.8	56.4	56.1	54.0	55.3	55.1	56.2	55.8	54.3
	(A) w/ Pop. thresh.	1.1	1.1	1.6	2.4	0.5	0.1	0.1	0.1	0.2	0.1
	(B) w/ Pop. thresh.	60.4	58.6	57.6	56.2	78.3	43.0	43.1	45.2	42.9	58.7

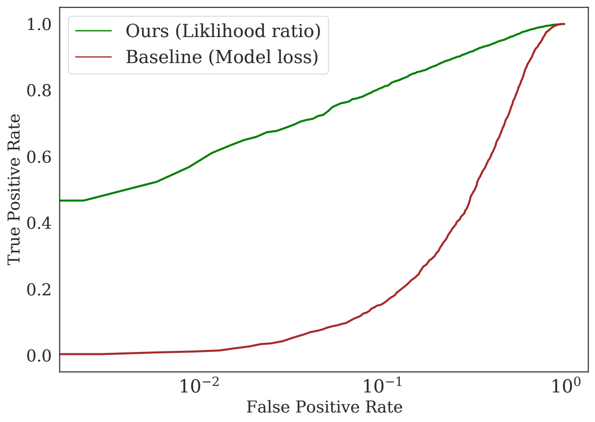


Figure 5: The ROC curve of sample-level attack, with logarithmic scale x-axis. performed on Clinical-BERT using MIMIC samples as non-members. Green line shows our attack and the red line shows the baseline loss-based attack. The blue dashed line shows AUC=0.5 (random uess). We find false positive and true positive rates over the target sample pool, which is comprised of members and non-members. This curve corresponds to Table 2 in the results, and is the zoomed-in version of Figure 4.

A.4 Extended Related Works

Since our work proposes an attack for quantifying leakage of masked language models (MLMs), based on the likelihood ratio, there are two lines of work that are related to ours: (1) work surrounding attacks/leakage on machine learning models (2) work on calculating sequence likelihood for MLMs. Prior work on measuring memorization and leakage in machine learning models, and specifically NLP models can itself be classified into two main categories: (1) membership inference attacks and (2) training data extraction attacks. Below we discuss each line of work in more detail.

Membership inference. Membership Inference Attacks (MIA) try to determine whether or not a target sample was used in training a target model (Shokri et al., 2017; Yeom et al., 2018). These attacks be seen as privacy risk analysis tools (Murakonda and Shokri, 2020; Nasr et al., 2021; Kandpal et al., 2022), which help reveal how much the model has memorized the individual samples in

Table 11: Sample-level attack results (with $\alpha = 10\%$ false positive rate used for thresholding) on the four ClinicalBERT models, plus the Base-b model (name insertion). Here we use "normalized energy" as a proxy for likelihood, instead of using the 15% masked energy formulation of Equation 2. We use PubMed-BERT as the reference model and the MIMIC data as non-members. Base++ (Large++) is same as Base (Large), but trained for more epochs. Base-b is the same as the Base model, but the training data was modified to prepend patient's first and last name to each sequence. This table studies effect of model size, training and sequence length on leakage.

		Short					Long				
	Non-members	Base	Base++	Large	Large++	Base-b	Base	Base++	Large	Large++	Base-b
AUC.	(A) Model loss	0.685	0.675	0.686	0.714	0.547	0.487	0.482	0.476	0.522	0.329
	(B) Ours	0.916	0.910	0.917	0.924	0.972	0.871	0.869	0.873	0.881	0.950
Prec.	(A) w/ μ thresh.	63.02	62.18	62.43	63.70	53.00	49.24	49.23	48.78	50.44	42.58
	(A) w/ Pop. thresh.	64.38	63.94	66.78	72.47	34.68	36.55	35.01	36.49	50.65	7.74
	(B) w/ Pop. thresh.	89.07	89.05	89.09	89.21	90.40	88.07	88.10	88.15	88.19	89.97
Rec.	(A) w/ μ thresh.	58.11	58.27	58.57	58.77	55.29	59.17	59.13	59.70	59.72	55.42
	(A) w/ Pop. thresh.	17.90	17.57	19.91	25.90	5.26	5.75	5.37	5.73	10.22	0.84
	(B) w/ Pop. thresh.	80.66	80.48	80.82	81.82	93.23	73.40	73.61	73.98	74.21	89.21

its training set, and what the risk of individual users is (Nasr et al., 2019; Long et al., 2017; Salem et al., 2018; Ye et al., 2021; Carlini et al., 2021a) A group of these attacks rely on behavior of shadow models (models trained on data similar to training, to mimic the target model) to determine the membership of given samples (Jayaraman et al., 2021; Shokri et al., 2017). In the shadow model training procedure the adversary trains a batch of models $m_1, m_2, ..., m_k$ as shadow models, with data from the target user. Then, it trains $m'_1, m'_2 ..., m'_k$ without the data from the target user and then tries to find some statistical disparity between these models (Mahloujifar et al., 2021). Shadow-based attacks have been mounted on NLP models as well: (Song and Shmatikov, 2018) mounts such an attack on LSTM-based text-generation models, (Mahloujifar et al., 2021) mounts one on word embedding, (Hisamoto et al., 2020) applies it to machine translation and more recently, (Shejwalkar et al., 2021) mounts it on transformer-based NLP classification models. Mounting such attacks is usually costly, as their success relies upon training multiple shadow models, and access to adequate shadow data for training such models.

Another group of MIAs relies solely on the loss value of the target sample, under the target model, and thresholds this loss to determine membership (Jagannatha et al., 2021; Yeom et al., 2018). Song and Raghunathan mount such an attack on word embedding, where they try to infer if given samples were used in training different embedding models. Jagannatha et al., which is the work closest to ours, uses a thresholding loss-based attack to infer membership on MLMs. Although our proposed attack is also a threshold-based one, it is different from prior work by: (a) applying likelihood ratio testing using a reference model and (b) calculating the likelihood through our energy function formulation. These two components cause our attack to have higher AUC, as shown in the results.

We refer the reader to the framework introduced by (Ye et al., 2021) that formalizes different membership inference attacks and compares their performance on benchmark ML tasks.

Training data extraction. Training data extraction quantifies the risk of extracting training data by probing a trained language model (Salem et al., 2020; Carlini et al., 2019; Zanella-Béguelin et al., 2020; Carlini et al., 2021b, 2022; Nakamura et al., 2020). The most prominent of such attacks, on NLP models is that of Carlini et al. (2021b), where they take more than half a million samples from different GPT-2 models, sift through the samples using a membership inference method to find samples that are more likely to have been memorized, and finally, once they have narrowed down the samples to 1800, they check the web to see if such samples might have been in the GPT-2 training set. They find that over 600 of those 1800 samples were verbatim training samples. Lehman et al. (2021) mount the same data extraction attack on MLMs, but their results are somehow inconclusive as to how much MLMs memorize samples, as only 4% of generated sentences with a patient's name also contain one of their true medical conditions. They also mount other type of attacks, where they try to extract a person's name given their disease, or disease given name, but in all their attacks, they only use signals from the target model and consistently find that a frequency-based baseline (i.e. one that would always guess the most frequent name/disease) is more successful.