INVESTIGATING PRIVACY CONCERNS AND MITIGATIONS FOR LANGUAGE MODELS IN HEALTHCARE

Victoria Smith
University College London
Institute of Child Health/Institute of Health Informatics
v.smith.20@ucl.ac.uk

Dan Schofield
NHS England
Transformation Directorate
daniel.schofield1@nhs.net

ABSTRACT

Language Models (LMs) have shown greatly enhanced performance in recent years, attributed to increased size (parameters) and extensive training data. This advancement has led to widespread interest and adoption, particularly in sectors which generate large amounts of text data, like Healthcare. However, a concerning revelation is that LMs tend to memorize their training data, and this memorization capability increases proportionally with the model's size. The memorized text sequences within these models have the potential to be directly leaked, posing a serious threat to data privacy. Various techniques have been developed to attack LMs and extract their training data, even when access to model parameters is restricted. This vulnerability applies to any LM trained on sensitive data released to a wider audience than those with access to that data directly. As these models continue to grow, this issue becomes increasingly critical. We present an initial exploration of privacy risks in healthcare LMs. We also explore privacy-preserving techniques applied to LMs before or after model training and evaluate their effectiveness with privacy attacks.

This work was completed as part of an NHS England Data Science PhD Internship project.

Keywords: Privacy Leakage, Language Models, Data Deduplication, Machine Unlearning, Machine Editing

Code for this Project: https://github.com/nhsengland/priv-lm-health

1 Introduction

Over recent years, there have been significant advancements in Language Models (LMs). This progress has resulted in larger, data-intensive models pre-trained on extensive datasets and then fine-tuned for specific tasks. These powerful LMs are being rapidly adopted in various fields with highly private data, including healthcare [1, 2, 3]. The public's awareness and usage of these models has also risen, evident in the widespread adoption of models like ChatGPT.

However, these models bring about several privacy risks. Studies have shown that Machine Learning (ML) models, including LMs, can inadvertently memorize and disclose information from their training data (termed training data leakage) [4, 5], jeopardizing the privacy of the data they were trained on. As LMs have grown, their ability to memorize training data has increased [6], leading to substantial privacy concerns. Worse, out-of-distribution data points, often posing the most significant privacy threats, are the most vulnerable to memorisation [7]. When trained on sensitive data, if LMs are subsequently accessible to users who lack direct access to the original training data, there is potential for this sensitive information to be leaked by the model. This is a concern even if the user has no malicious intent.

As LMs expand in size and usage within the healthcare domain increases, it is crucial to comprehend and address these privacy risks promptly. This document presents the technical report for the initial work investigating privacy concerns and mitigations in healthcare LMs at NHS England. This work has an accompanying repository under the name `priv-lm-health` (Investigating Privacy Concerns and Mitigations in Healthcare LMs), which is available on GitHub at `https://github.com/nhsengland/priv-lm-health` and contains code to reproduce the results contained below. Due to the potentially harmful use of any released attack code, we have not released this widely. However, please do contact us if you wish to use this in your experiments, and we are happy to share the code.

2 Language Models

This section gives an overview of LMs, the common architectures used, the training process and current trends in Language Modelling.

2.1 What are Language Models?

LMs are crucial in contemporary Natural Language Processing (NLP) workflows. They focus on understanding the likelihood of tokens[1], based on their surrounding context in a text sequence. These models are predominantly trained on extensive text datasets using self-supervised language modelling techniques. One common approach involves predicting subsequent tokens in a sequence based on a series of unannotated tokens.

These models, often called "foundation models" [8], have versatile applications in downstream NLP tasks like Natural Language Understanding and Generation. Natural language understanding encompasses tasks such as document classification, named entity recognition and normalization, sentiment analysis, and information extraction. On the other hand, natural language generation tasks involve predicting the next tokens e.g., text summarization and machine translation.

2.2 Architectures

Transformer-based architectures [9] are the current State-of-the-art behind large LMs. Transformers process the entire input sequence simultaneously and use an attention mechanism which enables targeted focus on particular tokens within the sequence. This capability allows the models to capture internal dependencies across long sequences. Depending on the specific use case, one can choose from three types of transformer-based LLM architectures: (i) encoder-decoder models, (ii) encoder models, and (iii) decoder models.

Encoder-decoder models In this setup, an encoder transforms an input sequence into a series of continuous representations. These representations are then processed by the decoder, which generates an output sequence of tokens, one at a time, by determining the most likely next token. This encoder-decoder framework is versatile and capable of handling sequence-to-sequence tasks with varying input and output sequence lengths, making it particularly suitable for machine translation and text summarization. Notable examples of this architecture include T5 [10] and BART [11].

Encoder models exclusively utilize the encoder component of the transformer and are focused on generating contextual representations for input tokens. Input sequences are transformed into feature vectors, incorporating contextual

[1] Small meaningful units of text, which can be words, sub-words or characters, used as a functional semantic unit for processing text.

information from the entire sequence (i.e., bidirectional information). These token feature representations find utility in various downstream NLP tasks, such as sequence classification (e.g., document classification and sentiment analysis), named entity recognition [12], relation extraction [13], and question answering [12]. Typically, a fully connected layer is added as the final layer of the model to facilitate these downstream tasks. Prominent examples of encoder models include BERT [12] and RoBERTa [14].

Decoder models exclusively utilize the decoder component of the transformer. The decoder processes a sequence of tokens as input, generating a feature vector. The feature vector is used to calculate the probability distribution for the next token in the sequence. Each feature vector in this context is informed solely by the current token and the tokens that precede it. This unidirectional nature makes decoder models well-suited for natural language generation tasks. The output token is auto-regressive, reusing past outputs as inputs for subsequent steps. Noteworthy examples of decoder models include GPT-2 [15] and GPT-3 [16].

2.3 Training

Previously, classical LMs (e.g. recurrent and convolutional neural network-based LMs) were trained in a supervised manner for specific tasks using labelled text datasets. The introduction of self-supervised tasks, pioneered by static word embedding models [17], marked a shift. Notably, the development of the Transformer architecture has ushered in a new era of very large LMs with billions to trillions of parameters and a two-phase training approach— pre-training and fine-tuning:

Pre-training: Transformers are typically pre-trained on a general language modelling task, utilizing extensive, minimally-curated unlabeled text corpora, often scraped from the internet [10, 18, 16]. Pre-training encoder models involve randomly masking tokens in the input text sequence, with the model learning to recover these masked tokens in the output. For instance, BERT [12] is trained on texts from BooksCorpus (800M words)[19] and English Wikipedia (2,500M words)[12], predicting masked tokens and next sentence sequences. Pre-training the decoder model involves computing the probability distribution of the next token. For example, GPT-3 is trained on English Wikipedia, Common Crawl, and Books 1 & 2 Corpora, using approximately 45TB of text [16]. Encoder-decoder Models can be pre-trained separately or unified with shared parameters [20, 11].

Fine-Tuning: Post pre-training, models are fine-tuned for specific tasks using smaller labelled task-specific datasets, which are more likely to be private [21, 22].

2.4 Use-cases in Healthcare

With the rapid increase in the size of Language Models, their utility has also increased, making them well-suited to numerous NLP tasks. In healthcare, there is a wealth of unstructured text data, making the field particularly primed to benefit from modern LMs, which can help structure the data and unlock unseen insights. However, it is also notable that this data is often highly sensitive. The following lists some uses of LMs in Healthcare, although this is by no means comprehensive:

- Analysis of Clinical Notes e.g. to build patient cohorts for analysis, to enhance diagnostics, to match patients to clinical trials:
  - Named Entity Recognition to identify key concepts in the text e.g. conditions, medications, symptoms, procedures, adverse events.
  - Error Detection
  - Conversion of concepts in e.g. conditions to medical codes (e.g. ICD) etc.
  - Entity Normalisation to collect different expressions of a single concept under one entity e.g. conditions, medications, symptoms.
  - Relation Extraction to connect related entities together e.g. medications related to what condition.
  - Temporal Relations of events e.g. time doses of a drug are given.
  - Causal Relations e.g. causality between symptoms and treatment
- Trend analysis of Patient Forums (can be closed forums) - N.B. NHS UK currently uses LMs (BERT) in production to analyse free-text feedback on hospitals and GPs.
- De-identification of Clinical Notes.
- Prediction of disease risks and outcomes from clinical notes e.g. survival, time-to-discharge, conditions, medications, timelines, etc.

- To create representations for other downstream tasks e.g. building knowledge graphs and analysis of text similarity compared to other sources (e.g. to find similar patients).
- Conversational e.g. for Communication with patients (e.g. triaging, offering assistance) or staff (e.g. diagnosis aid, medical QA).
- Text Generation:
  - Personalized treatment plans.
  - Discharge summaries from notes.
  - Personalized educational resources based on patient's conditions.
- Text-to-text:
  - Summarisation of medical notes.
  - Converting audio to text (e.g. home safety monitoring in dementia (v.private data)).
  - Converting image or video to text (surgery tracking, describing/explaining medical images).

2.5 Summary

LMs are essential in contemporary NLP workflows. The training regimen of modern LMs entails an initial pre-training phase on extensive, unannotated text corpora, succeeded by fine-tuning on more specific, labelled datasets tailored for particular tasks. The paradigm shift from conventional supervised training to self-supervised tasks signifies a transformative era marked by the emergence of large LMs encompassing billions to trillions of parameters. The enhanced functionality has driven the widespread adoption of LMs in Healthcare, owing to the large amounts of unstructured text data generated within healthcare pathways. However, with this heightened adoption, it becomes critical to comprehend the inherent privacy risks associated with these LMs, given the sensitive nature of Healthcare data.

3 Privacy Concerns & Mitigations for Language Models

This section gives an overview of the privacy concerns in LMs, privacy attacks which can be staged against LMs and defensive methods to protect against these.

3.1 LMs Memorize Training Data

Studies have demonstrated that LMs memorize certain aspects of their training data, and this memorized information to be extracted verbatim when the LMs are prompted in certain ways, a phenomenon referred to as *training data leakage* [4, 5, 7, 23, 24, 25, 6, 26]. This leakage can violate the privacy assumptions under which datasets were collected and can make diverse information more easily searchable. As LMs have grown, the increasing over-parameterization of models has made them more susceptible to training data memorization [26, 6].

Several studies have delved into this memorization phenomenon in LMs. Personal details like URLs and phone numbers or artificially inserted "secrets" in training data can be extracted solely from the trained model [7, 23, 24, 25]. For instance, GPT-2 was observed to memorize 600 out of 40GB of training examples [7], and it can reproduce passages from the training data exceeding 1,000 words [26]. More recently, experiments with the large GPT-J LM (6 billion parameters) revealed that it memorizes at least 1% of the training data when prompted with training data prefixes [6].

Memorization has been found to correlate with (1) the size of the model (number of parameters)[6] and (2) duplicated sequences in the training data[27, 28, 6]. The extent of memorized training data also increases with the number of tokens in the prompt given to the LM [6]. In practical terms, this implies that some memorization becomes noticeable only when the LM is prompted with sufficiently lengthy context strings. Recent examples encountered by LMs during training are the most likely to be memorized, raising concerns, especially for LMs fine-tuned on private datasets, as these recent examples are the most recently observed by the model [29].

3.2 Privacy Attacks LMs

In this section, different privacy attacks that can be staged against LMs are outlined. It is important to understand the potential objectives of an attacker and their knowledge of the model and data.

3.2.1 Attacker Goals and Knowledge

Privacy attacks on LMs are focused on extracting information regarding the training data or the model itself. These attacker objectives can be categorized into four primary groups. In the below sections, these attacks are described briefly. For more detail on the different attack types and research in these areas, please refer to [30].

- **Membership inference attacks.** These attacks determine whether a specific text data instance was included in the training data of the targeted LM.
- **Model inversion (attribute inference) attacks.** These attacks deduce aspects of the original text data, particularly sensitive attributes, used in training the LM.
- **Data extraction attacks.** These attacks aim to reconstruct text sequences verbatim, without specific attribute inference, that were used in training the LM..
- **Model extraction attacks.** These attacks involve reconstructing the parameters of the LM from the LM itself.

Privacy attacks on models can be conducted at two distinct levels of adversarial access:

- **Black-box access** refers to where the attacker possesses only input-output access to the LM. The attacker's knowledge is restricted to the probabilities of arbitrary sequences and predictions of the next token.
- **White-box access**, on the other hand, assumes that the attacker possesses certain knowledge about the LM, such as its architecture, parameters, training data distribution, or gradients.

While white-box access is less frequently encountered in practical situations, there are still conceivable scenarios where it might happen. For instance, this could occur if: (i) the attacker is an insider within an organization with access to the trained model; (ii) the entire model is made publicly available along with knowledge of the training data distribution; or (iii) in a federated training setup where the attacker can intercept communication between the central- and client-server.

3.2.2 Membership Inference Attacks

(MIAs) try to deduce whether a specific data point was part of a target-model's training dataset, as described in Shokri *et al.* [4]. Such attacks can result in various privacy breaches, for instance, being able to discern that a text sequence generated by a Clinical LMs (trained on Electronic Health Records) originating from the training data can disclose sensitive patient information, violating their rights [31]. MIAs have been executed in black box scenarios against supervised LMs[32, 33, 34], static word embedding Language Models [35], as well as pre-trained [36, 31], fine-tuned [37], and compressed [31] LMs.

Loss-based MIAs. There are many methods for mounting MIAs; one of the most common is loss-based MIAs. At the simplest level, loss-based attacks use the loss of the target data instance under the target model to predict membership. A specific threshold, t, is utilized on the loss value to ascertain membership status. This threshold can be determined by: (1) Using the mean of the training samples loss [31] or (2) calculating the loss on a population set of samples (samples that were not used in training but are similar to training data) and then selecting the threshold that would result in a 10% false positive rate on that population [36]. If $loss \leq t$, then the adversary rejects the null hypothesis, assuming the target is a member of the training data. Otherwise, the adversary fails to reject the null hypothesis.

Reference Models. Generally, the most successful MIAs use *reference models* [4]. This refers to a second model trained on a dataset similar to the training data of the target model. The perplexity of the target LM when predicting a given sequence is used to identify the membership of the training data. A small perplexity on the target LM and a relatively large perplexity on the reference LM indicates the sequence is "surprising" to the reference LM but not to the target LM and was, therefore, likely a member of the training data. So, the reference model filters out common samples, which will also have high confidence scores from the reference model.

Likelihood-ratio (LR) -based MIAs. These attacks have been demonstrated to be a stronger form of MIAs [36]. The Likelihood Ratio (LR) test compares the log-likelihood ratio statistic $L(s)$ with a threshold t. This needs some adaptation for Masked LMs as these models do not explicitly define an easy-to-compute probability distribution over natural language sequences, unlike generative LMs. To estimate the likelihood ratio for an input sequence, 15% of tokens can be masked in the sequence at random, k times to generate a set of masking patterns for the sequence. Each of these combinations of masking patterns is then passed in a forward pass through the model to calculate the output scores, which are then combined to give the probability distribution over this sequence [36]. The choice of masking 15% of tokens is made to reflect the typical masked language model training process. However, equally possible but more computationally expensive is to mask every token in the sequence one at a time [38]. Another option is to mask

tokens from the same word simultaneously, which has been suggested to give better results (but again is computationally more expensive) [39]. Please refer to [38, 39] for further information on this topic. Mireshghallah *et al*. [36] carry out a LR-MIA using ClinicalBERT (trained on MIMIC-III) as the target model and PubMedBERT as the reference model, achieving a high AUC of 90% in terms of being able to distinguish members and non-members.

3.2.3 Model Inversion and Attribute Inference Attacks

Model inversion or attribute inference attacks aim to deduce sensitive attributes of a partially known data record used in training a target model [40, 41]. These attacks assume that an attacker possesses information about non-sensitive attributes of the data record and has access to the output of the trained model. Specifically, when a data record has a missing attribute with t possible values, the attacker constructs t different input vectors and feeds them into the target model. The model's perplexity is then used to select the input most likely to be a member of the target model's training dataset [5]. For LMs, model inversion and attribute inference attacks have primarily been conducted in white-box settings. For instance, Abdalla et al.[42] demonstrate how they can link name tokens to specific diagnoses using static word embeddings trained on medical consultation notes. Lehman et al.[43] probe BERT, trained on electronic health records, using a fill-in-the-blank template to recover patient names and their associated conditions from clinical datasets, although they find this attack is weak, achieving correct associations in only 4% of cases.

3.2.4 Data Extraction Attacks

Data extraction attacks are designed to extract training data indiscriminately. These attacks have been conducted on supervised LMs[44] and pre-trained[23, 44] and fine-tuned [37] Transformer models. In instances where only black-box access is available, it has been demonstrated that hundreds of text sequences containing names, addresses, and phone numbers can be extracted from the pre-training data of GPT-2. Remarkably, this extraction can occur even when these sequences are found in just a single training document [23]. To execute this attack, the researchers generated a substantial amount of data by unconditionally sampling from GPT-2. They then filtered out generated samples unlikely to contain memorized text by employing a LR-MIA. Similar data extraction attacks under the same assumptions have also been applied to fine-tuned LMs, which were especially vulnerable to training data leakage when only the prediction layer of the model was tuned [37].

3.2.5 Model Extraction Attacks

Model Extraction Attacks, also known as model stealing, are attempts to replicate the functionality of a target model by constructing a model with comparable predictive performance. This practical threat has been demonstrated in cases where LMs accessible via APIs have been vulnerable to model extraction [45, 46, 47]. Chen et al. [47] employed outputs from well-crafted queries on a BERT-based API to fine-tune a pre-trained BERT model, creating a replicated model. They demonstrated that this extracted model is susceptible to information leakage through white-box attacks, revealing sensitive attributes of the training data used for the target API model. Krishna et al. [45] showed that it is possible to extract fine-tuned BERT-based LMs without knowledge of the training dataset and even when queries consist of randomly sampled sequences of words. Furthermore, their work highlighted that matching pre-training architectures is not essential; attackers can leverage similar or more powerful LMs for successful model extraction.

3.3 Privacy Mitigations for LMs

Extensive research has been conducted to address the leakage of sensitive user information from training data. Three primary approaches have been explored:

- Methods for data preprocessing, like sanitizing (eliminating private information) or deduplicating training data.
- Training strategies, such as utilizing privacy-preserving learning algorithms (e.g., differentially private training).
- Post-training techniques, such as Machine Unlearning or Editing.

3.3.1 Pre-processing Approaches

Data sanitization, aims to remove all sensitive information from data before model aim to eliminate all sensitive information before model training [48, 49]. These techniques are widely applied in healthcare to remove Personally Identifiable Information (PII). However, for large text datasets, identifying and removing specific sensitive sequences from the text is only possible through automated methods such as pattern-based parsers or classification algorithms, leading to potentially imperfect outcomes. Formally defining private information in Natural Language is inherently

complex [16], making it challenging to design a sanitization method to guarantee the removal of all potentially sensitive sequences. Data sanitization approaches are effective when sensitive information follows a context-independent, consistent format (e.g., national security numbers, email addresses, etc.). The effectiveness of data sanitization in preserving privacy cannot be precisely measured or guaranteed, indicating that it should not be relied upon as the sole privacy-preserving measure for LMs.

Data Deduplication. Training datasets for LMs are typically deduplicated at the document level e.g. removing repeated webpages. However, text sequences are often duplicated both within and across different documents. Recent studies have shown that eliminating duplicate sequences from the training data, utilizing a suffix array-based algorithm, results in GPT-2 LMs generating approximately 10x less training data [28]. Expanding on this, empirical evidence indicates that the likelihood of GPT-Neo LM generating exact sequences from the training data increases in proportion to the presence of duplicates in the training data [6]. Furthermore, removing exact duplicate sequences from the training data safeguards GPT and LSTM LMs from extraction attacks without compromising model performance [27]. This suggests data deduplication is an effective measure to prevent training data leakage from large LMs. Importantly, data deduplication aims to reduce memorization on average and cannot guarantee the prevention of memorization of a specific training example.

3.3.2 Training Approaches

Differential Privacy. A model is considered differentially private (DP) if its outputs on neighbouring datasets that differ by only one record are statistically indistinguishable. Specifically, an algorithm A is (ε, δ)-DP if for any neighbouring datasets, D and D', that differ in only one record and all $S \in Range(A)$:

$$Pr[A(D)\ S] \leq e^{\varepsilon} Pr[A(D') \in S] + \delta \tag{1}$$

Here, the privacy parameter ε sets an upper limit on the potential privacy leakage in the worst-case scenario. A smaller ε indicates stronger privacy protection. The parameter δ is in the order of the inverse of the dataset's size, so it is typically very small.

DP Stochastic Gradient Descent (DP-SGD) represents the standard approach for training machine learning models with DP [50, 51, 52]. In each training iteration, DP-SGD introduces two modifications to vanilla SGD: (1) the gradients for individual examples are clipped to a fixed norm, C, to limit the influence of individual training examples on model updates, and (2) calibrated Gaussian noise, proportional to C, is added to the aggregated clipped gradients. This noisy gradient is then utilized to update the model parameters. The amount of noise added, relative to the clipping norm, determines the strictness of the upper limit ε on privacy loss that can be ensured.

DP for LMs. Achieving robust privacy guarantees in transformer-based LMs comes at a cost to utility and efficiency [53]. DP-SGD often results in a notable reduction in accuracy due to the application of gradient clipping and noise addition at the level of individual training examples. Moreover, computing per-example gradients in DP-SGD incurs substantial memory and computational overhead, posing significant challenges for transformer-based language models, which typically consist of hundreds of millions of parameters. Parameter-efficient methods have been utilized to fine-tune BERT and GPT-based LMs under differential privacy (DP) [54]. These techniques, including Low-Rank Adaptation, Adapters, and Compactors, aim to decrease the number of adjustable parameters in the model. This approach is grounded in the idea that achieving high performance does not necessarily require the full model parameters. Interestingly, the study's authors also discovered that larger LMs attain the highest accuracy when trained with DP.

3.3.3 Post-training Approaches

Machine Unlearning. Unlearning comprises a set of techniques to remove the influence of specific training examples from the weights of a trained model without retraining the model from scratch. This could be used, for example, when somebody wishes to practise their Right-to-be-Forgotten, removing private data from the training data after model training. This is particularly significant for LMs as retraining is highly costly and impractical. There is a growing body of work on unlearning in machine learning models, with a NeurIPs competition on this topic for Image Models[2]. However, this area is relatively new for LMs, and the literature is restricted. An early example by Jang et al.[55] utilized gradient ascent, adjusting the direction during language modelling to maximize the loss function, to "unlearn" specific examples (the "forget set") from the original training data. They showed this method effectively protects the "forget set" from extraction attacks [23], with minimal degradation to performance for GPT-Neo. In recent work, a new unlearning method was introduced to teach Llama2-7b, Meta's generative Language Model, to forget content from the

[2] `https://unlearning-challenge.github.io/`

Harry Potter books [56]. This technique required only one GPU hour of fine-tuning, making it significantly faster than the approximately 184,000 hours needed for retraining the model. The approach involved using a reinforced model that underwent additional training on the target data to identify tokens closely linked to the unlearning objective by comparing its logits with those of a baseline model. Unique expressions in the target data were replaced with generic equivalents, and the model's own predictions were used to create alternative labels for each token, simulating predictions from a model not trained on the target data. The model was then fine-tuned using these alternative labels, successfully eliminating Harry Potter-related content from memory without impacting performance on common Language Model benchmarks.

Machine Editing aims to update a model's behaviour concerning a specific edit descriptor by updating, erasing or inserting knowledge. Machine Editing techniques overlap with Machine Unlearning but are more focused on changing or erasing a specific "fact" (e.g. the president of the US is Joe Biden) instead of forgetting certain training data instances. There have been numerous works on model editing in large LMs [57, 58, 59, 60] using a variety of methodologies which can be grouped into the following: (1) Memory-based techniques, which do not change model parameters, using another model or another feedforward layer at the end of the network to change the behaviour e.g. [61, 59]; (2) Locate-then-edit techniques, which attempt to identify which parameters in the network store specific knowledge and modify them to remove or alter the model output for this knowledge e.g. [60, 58, 62]; (3) Meta-learning approaches which employ a hyper network to learn the necessary parameters for editing the model e.g. [57, 63].

Machine Editing and Unlearning techniques are effective tools for modifying or erasing information post-training, rendering them highly applicable in real-world scenarios. Nevertheless, it's crucial to understand that these methods, on their own, lack formal privacy assurances. The evaluation of Editing and Unlearning Algorithms is inconsistent across the literature, and the methods have often been applied to different LM types, making it challenging to compare their effectiveness. Further research is vital to incorporate them as complementary measures alongside other methods that establish privacy safeguards during the initial stages of the training process.

3.4 Summary

Overall, training data leakage, where LMs memorize specific aspects of their training data, is a growing concern for large LMs in domains with sensitive data, such as Healthcare. Training data leakage violates privacy assumptions on the dataset and can lead to diverse information being easily searchable. Various studies demonstrate memorization in LMs, revealing their ability to reproduce passages and extract personal details from training data. Privacy attacks aim to extract information about the training data or the model itself and can be staged successfully even when the attacker only has input-output access to the LM. There are several mitigation strategies which have been developed to reduce data leakage and target phases of data pre-processing, LM training and fully trained LMs. However, further research is needed to establish comprehensive recommendations and practices for LM development to safeguard the privacy of sensitive training data.

4 Methodology

4.1 Datasets

MIMIC-III. The Medical Information Mart for Intensive Care III (MIMIC-III) was used to conduct our experiments [64]. The dataset was downloaded from the Physionet page [3] [43]. Please note this dataset requires a license to download. Please refer to Lehman *et al.* [43] for a detailed explanation of the pre-processing this dataset underwent. In short: (1) all notes are removed except for those categorized as 'Physician', 'Nursing', 'Nursing/Others', or 'Discharge Summary'; (2) the data is deidentified using a combination of regular expressions and human oversight to remove personal health information (PHI); (3) to simulate the existence of PHI in MIMIC-III the data is pseudo-reidentified using randomly sampled names sampled from US Census data. This processed MIMIC-III dataset comprises 1,247,291 electronic health records (EHR) of 46,520 patients. Subsections of this dataset are used to train our LMs and as members and non-members for our Membership Inference attacks.

i2b2. This dataset was curated for the i2b2 de-identification of protected health information (PHI) challenge in 2014 [65]. We specifically use the De-identification and Heart Disease Risk Factors Challenge Downloads Gold Training Sets 1 & 2 and Testing Set. This dataset was downloaded from the DBMI Data Portal [4]. Please note this dataset requires a license to download. We only use this dataset as a second non-member dataset (as it is from a similar domain to MIMIC-III) for our Membership Inference attacks (it is not included in the training dataset for our models).

4.2 Data Preprocessing

Near Duplication with Minhashing aims to remove full training examples or documents with high token n-gram overlap. The technique removes all but one near duplicate. Logically, training on near duplicate examples results in the model seeing the same long text sequences multiple times per epoch, making it more likely to memorize them. The methodology is as follows: (1) Tokenization (also called shingling)- documents are space tokenized; (2) Minhashing (also called fingerprinting)- each consecutive n-gram is hashed using tabulation hashing. So, a set of these hashes is the signature for the document. For each element in a document's signature, the element is hashed using k other hash functions. The minimum hashed element for each k hash function is stored; (3) Duplicate Removal- the minimum hashes are then partitioned into r buckets, with b hashes per bucket. These b hashes are augmented into a single value; then, if two documents have the same value in at least one bucket, they are marked as a potential match. For document pairs identified as potential matches, if the Jaccard index (Equation 2) between the two documents i and j is above 0.8, the edit similarity (Equation 3) between token sequences x_i and x_j is computed, and if this is above 0.8, the document pairs are identified as duplicates, and one is removed.

$$Pr(d_i, d_j | Jaccard(d_i, d_j) = s_{i,j}) = 1 - (1 - s_{i,j}^b)^r \tag{2}$$

$$EditSim(x_i, x_j) = 1 - \frac{EditDistance(x_i, x_j)}{max(|x_i|, |x_j|)} \tag{3}$$

Exact Substring Deduplication with Suffix Arrays looks for text sequences of a specified minimum length that overlap highly in the training data and removes all but one of these. The logic behind this method is that due to the diversity of possibilities in human language, it is rare for the same idea to be expressed identically in multiple documents unless one expression is derived from the other or both are quoting from a shared source. The method is as follows: (1) The text is tokenized with GPT2 byte pair encoding (BPE) Tokenize; (2) All the examples of the entire dataset D are concatenated into a giant sequence S; (3) A Suffix Array A is constructed from S using the SA-IS Algorithm and ordered lexicographically (e.g. suffixes of the sequence "banana" are ("banana", "anana", "nana" "ana", "na", "a", so the suffix array is the sequence (6 4 2 1 5 3) when ordered lexicographically); (4) Repeated sequences occur adjacent to each other in the suffix array A. This enables efficient identification of duplicated training examples by linearly scanning the suffix array from beginning to end and looking for sequences $A_i ... A_{i+1}$ that share a common prefix of at least some threshold length (e.g. 100 tokens) and when this common prefix is found, remove it from all but one sequence.

Semantic Deduplication using Embeddings aims to find sentences with high semantic similarity and removes all but one of these. This works by leveraging pre-trained models to identify and remove data pairs which are semantically

[3] `https://physionet.org/content/clinical-bert-mimic-notes/1.0.0/`

[4] `https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/`

similar. One study found that upon blocking LMs from generating token sequences from the training data, the LM would instead get around this by generating semantically similar sequences, meaning it could still leak private information in a reworded format [66]. Thus, removing semantically similar phrases from the text could reduce overall memorization of sequences as the LM will see fewer things with similar meanings. Abbas *et al.* [67] found that semantic deduplication removed large amounts of data and sped up pretraining time whilst preserving performance. We performed some initial experiments on MIMIC-III to explore different approaches to semantic deduplication.

Unfortunately, we ran out of time to conduct the full range of experiments with this technique (pre-training, fine-tuning, MIAs). As there was no repository to apply Semantic Deduplication out-the-box, we experimented with different setups. We explored topic modelling to visualise the semantic clusters after embedding and clustering to sanity-check these. The process in topic modelling is as follows: (1) Embedding - embed the documents; (2) Reduce Dimensionality of embeddings; (3) Cluster the reduced embeddings; (4) Tokenize the clusters and weight the tokens; (5) Display the top n tokens for each cluster. We split the text from the patient notes in MIMIC-III into sentences and explored the clusters generated by both Top2Vec and BERTopic. Top2Vec uses distilBERT, whereas BERTopic allows you to plug in any pre-trained LM, so we used ClinicalBERT. For clustering, Top2Vec uses K-means, which is a distance-based clustering algorithm, whereas BERTopic uses HDBSCAN as a default, which is a Hierarchical clustering algorithm which uses stability to establish clusters.

4.3 Language Modelling

4.3.1 Continued Pre-training RoBERTa

Data. Pre-training was carried out using a 50% split of the MIMIC-III Notes data described above. This was due to time constraints and to reserve some data for Membership Inference Attacks - see below. Three separate variations of this 50% MIMIC-III split with different pre-processing methods (see Table 1) were used to train three separate models.

Table 1: Summary of pre-training datasets.

Dataset	Deduplication Technique
Mimic-III-50%	None
Mimic-III-50%	Near Duplication with Minhashing
Mimic-III-50%	Exact Substring Deduplication with Suffix Arrays

Architecture. We used a masked language modelling architecture, RoBERTa. This choice was driven by the fact we wanted to start our Membership Inference Attacks on this type of LM and for the ease of training. RoBERTa is a modified version of the BERT model, featuring adjustments in its pretraining configuration and optimizations to enhance the original selection of training hyperparameters. Studies have demonstrated its superior performance over BERT in certain contexts, and it is quicker to train.

Tokenization. Before text can be fed into any model, it must be split into small meaningful units called tokens, known as tokenization. Tokenization involves building a vocabulary from the training corpus and assigning a unique numeric ID to each token. Once a tokenizer is trained, the vocabulary is set, and any words not previously encountered by the tokenizer are assigned the same token ID representing “unknown". At the simplest level, tokenization involves splitting text on whitespace into individual words where each is a token. However, it is typical to encounter many new whole words when tokenizing a new text corpus, resulting in high numbers of “out-of-vocabulary" terms, meaning information will be lost from the text sequence. The preferred method is to use sub-words, which allows us to represent the text using the least number of tokens whilst avoiding “out-of-vocabulary" terms. Common words will be represented as whole words in the vocabulary, and rare words will be split into commonly occurring sub-words. Upon encountering a new “out-of-vocabulary" word, the sub-word tokenizer can represent this as a collection of sub-words from the vocabulary. RoBERTa employs one such sub-word tokenizer algorithm, Byte-Pair Encoding (BPE). This eliminates the occurrence of unknown tokens by iteratively splitting unknown words until recognizable sub-word pieces are identified.

Continued Pre-Training Set-Up. We started with RoBERTa-base and continued pre-training this model with our data. To improve training speed, we investigated the cramming GitHub repository [5] but found, unfortunately, this was only set up for pre-training from scratch and did not support continued pre-training. Based on our investigations, we recommend this repository for anyone wanting to pre-train from scratch. Instead, we adapted the NHS ELM4PSIR

[5] `https://github.com/JonasGeiping/cramming`

GitHub repository [6] for pre-training. We used the same hyperparameters implemented in this repository and did not try different variations. We only trained for a single epoch due to time constraints, which equated to ~17 hours of training.

Masked Language Modeling Objective. Masked Language Models are so-called as they are trained with Masked Language Modeling (MLM) as the objective. This involves randomly masking or replacing 15% of the input tokens with special <mask> tokens. The cross-entropy loss is then calculated based on these masked token positions, akin to the model filling in the gaps with predicted tokens. The original MLM, BERT, was trained with a second objective, next sentence prediction, where tokens of the second sentence are masked, and the model is asked to predict these. However, RoBERTa only used MLM as the objective. Another alteration in RoBERTa is dynamic masking. During the batching process, a certain percentage of each training sequence is randomly masked. BERT instead uses static masking, where masked datasets are created before training.

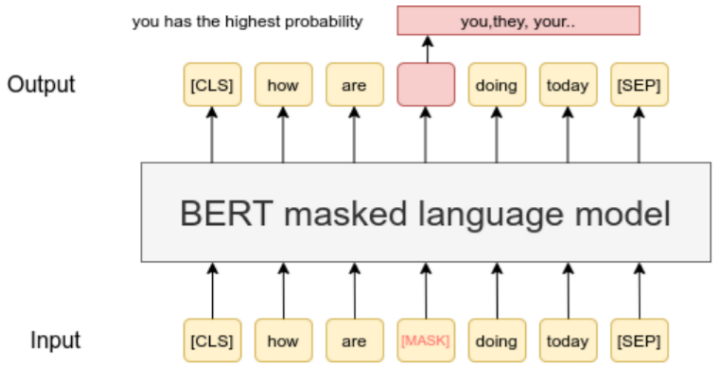


Figure 1: Example of Masked Language Modelling objective used by Masked LMs e.g. RoBERTa/BERT. A token in the input is replaced with the [MASK] token, and the LM's objective is to predict the true word in that masked position correctly. Figure taken from [68].

4.3.2 Continued Pre-training nano-GPT

We later continued pre-training nano-GPT models with the same split of data with which we trained the RoBERTa models, taking inspiration from the nano-GPT GitHub [7]. Due to time constraints, the following experiments were only performed with our trained RoBERTa models. However, these GPT models are available for future experimentation on this topic.

4.3.3 Fine-tuning

Fine-tuning involves training a pre-trained LM for a specific downstream task on a task-specific dataset by either (1) incorporating a task-specific layer or layers and updating the parameters of this/these whilst freezing the parameters in the rest of the network; (2) Updating the parameters of the entire pre-trained LM. In our, we used approach (1), freezing the pre-trained LM and adding a task-specific layer. Our downstream tasks are all document classification tasks. For document classification, the downstream task head uses a multi-layer perceptron denoted as fMLP(·). This MLP takes the sentence embeddings pooled from the pre-trained LM output as input and generates an n-dimensional vector, where 'n' represents the number of classes. In this context, given an input text x, the raw input is first processed by the pre-trained LM to obtain m-dimensional embeddings for each token. Subsequently, a pooling operation, such as the mean, is applied to all token embeddings to produce a singular sentence embedding h(x) with the same dimension m. This h(x) embedding is then fed into the MLP block in a standard feed-forward manner to obtain probabilities across n classes using a softmax operation.

30-day Readmission Prediction. The main downstream task which we fine-tuned and evaluated our models was the 30-day Readmission prediction tasks for MIMIC-III from the original ClinicalBERT paper [69]. The purpose of running this downstream task was as a sanity check on our pre-training, to ensure we could achieve similar results with our "Clinical-RoBERTa" to the original ClinicalBERT paper [69]. We prepared the fine-tuning data as in the

[6] https://github.com/nhsx/ELM4PSIR

[7] https://github.com/karpathy/nanoGPT

original ClinicalBERT paper [69], using their associated GitHub repository [8]. We then fine-tuned our three pre-trained RoBERTa models using the Trainer from Hugging Face [9]. The hyperparameters we used are detailed in Table 2.

Table 2: Fine-tuning Hyperparameters

Hyperparameter	Readmission	GLUE
Learning Rate	1×10^{-5}	2×10^{-5}
Train Batch Size	16	16
Gradient Accumulation	2	2
Weight Decay	0.01	0.01
Training Epochs	3	5
Warmup Steps	100	0

GLUE Tasks. We also fine-tuned and evaluated the models on a selection of tasks from the General Language Understanding Evaluation (GLUE) benchmark dataset [70]. The motivation for using this benchmark was another sanity check on our pre-trained models, which were initialised from RoBERTa-base, to probe problems like catastrophic forgetting [10] or over-fitting from our continued pre-training. The implementation of these was also with the Trainer from Hugging Face, as above.

4.4 Membership Inference Attacks

All MIAs were implemented with the likelihood ratio (LR) and loss-based methodology following [36].

4.4.1 Pre-training Data Deduplication.

In this experiment, we aimed to understand the effect of different deduplication methods on the privacy preservation of the pre-training data.

Data. Member, Non-Member and External datasets were curated for the MIAs. For each set, 1000 sentences were randomly selected without replacement from 90 randomly selected patients. We used the nltk library [11] to split patients' notes into sentences, then randomly selected sentences across the selected patients. The Member Set was selected from the 50% of MIMIC-III used for pre-training. The Non-Member set was selected from the held-out 50% split of MIMIC-III. Due to the high repetition of patient notes across MIMIC-III, patients who did not appear in the 50% training split were specifically selected to make up the Non-Member set to ensure no overlap in sentences. For the External dataset, patient notes from the i2b2 dataset were used.

Models. For our target models, we used our pre-trained Clinical-RoBERTa variations trained with different deduplication methods (see Table 1). For our reference models, we compared two different models: (1) RoBERTa-base-PM-Voc from Facebook [12], as this was estimated to be from a similar underlying distribution to MIMIC-III and therefore offer more specificity in the attack and; (2) RoBERTa-base, as a general domain model and as this formed the base of which our models were initialised. In reality, as many models are initiated from common base models, which are openly available, this is something an attacker is likely to have access to.

Threshold. For loss and LR-based MIAs, a test threshold t is selected to decide if a given sample is a member of the original training data. This threshold represents the tolerance of MIA error based on your member distribution (see Figure 2). The threshold is used to separate member and non-member model-output distributions. Ideally, the distributions of outputs from the target model when you feed it data points its seen before (members) are distinguishable from data it's never seen (non-members). This threshold value can then calculate the true positives, true negatives, false positives and false negatives for your member and non-member text sequences. For our MIA, we used a threshold at alpha=0.1 for the member distribution and used this to calculate the success rate of the attack.

Quantifying the Privacy Risk. The effectiveness of an MIA can be measured by analysing the relationship between the MIA's power (true positive rate) and its error (false positive rate). This is done by calculating the AUC-ROC score. Here, a high AUC-ROC score equates to higher power with lower errors, which signifies a greater privacy loss. To

[8] `https://github.com/kexinhuang12345/clinicalBERT`

[9] `https://huggingface.co/docs/transformers/training`

[10] Losing significant general domain knowledge when training or fine-tuning more general models on a specific domain

[11] `https://www.nltk.org/`

[12] `https://github.com/facebookresearch/bio-lm`

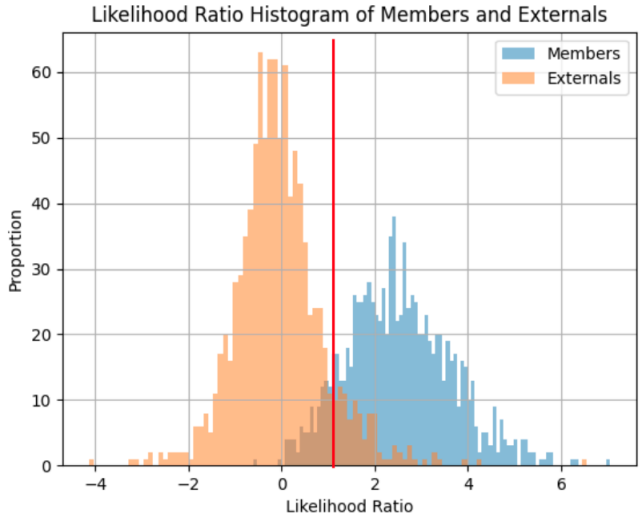


Figure 2: Example of Error Threshold on the Member Distribution at alpha=0.1. The threshold is represented by the red line.

calculate a threshold-independent AUC score of the overall privacy loss incurred by the MIA, the attack power is calculated for various threshold values (error values). This involves using values of $0 \leq \alpha \leq 1$ and calculating the true positive rate and false positive rate in each case to get the probabilities of each point being classified correctly under different threshold values. This can then calculate a threshold-independent AUC-ROC score for the MIA.

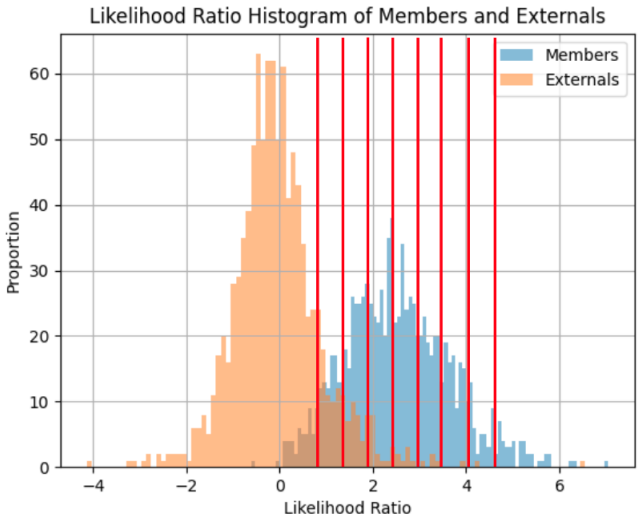


Figure 3: Example of Thesholds used to calculate Theshold-Independent AUC-ROC score. Thresholds are represented by red lines.

4.5 Post-training Approaches

4.5.1 Review of Model Editing & Unlearning Repositories

Table 3 includes all repositories we investigated for Editing and Unlearning methods for LMs. The EasyEdit [13] [71] repository has brought many of these repositories together and is well-documented and set up for plug-and-play.

[13] `https://github.com/zjunlp/EasyEdit`

However, it is only adapted to certain LMs (e.g. GPT2xl, GPT-J, GPT-Neo, T5, LLaMa, among others), which are not the LMs we have trained for our experiments, and so we were not able to use this out-the-box. We only included methods with an associated repository, and we assessed the difficulty of adapting this to our use case based on the documentation and code in the repository. It can be seen in the table that not all repositories have code available to adapt their method to all model types. Of the two ready-to-use MLMs, Knowledge Neurons was the only one with a well-documented repository. For this reason, we decided to experiment with Knowledge Neurons; see below.

Table 3: Summary of Information on Model Editing & Unlearning Repositories.

Approach	Maturity	MLM	De	En-De
Knowledge Neurons [72]	High	✓	✓	✓
Knowledge Editing [57]	Low	✓	✓	✓
ROME [58]	High		✓	
MEMIT [62]	High		✓	
MEND [63]	Low		✓	
Knowledge Unlearning [55]	High		✓	

MLM= Masked Language Model, De= Decoder LM, En-De = Encoder-Decoder LM. High Maturity = repository is well documented and readily usable with minimal changes, easy to adapt to data and models; Poor Maturity = little or confusing documentation, difficult to adapt code to own data and models.

4.5.2 Knowledge Neurons

Knowledge Neurons [72] is a method to find neurons associated with certain knowledge or facts within a LM network. Knowledge Neurons is a locate-then-edit technique which relies upon the idea that factual knowledge can be located within the network. The researchers start with a relational fact like "the capital of Ireland is Dublin" and initially use a Knowledge attribution technique to pinpoint the "knowledge neurons" that embody that fact. To measure the contribution of a neuron to factual expressions, they gradually change the weight of a neuron from 0 to the original value in the pre-trained model and observe the contribution to the output probability of the correct output. They do this process for several prompts about the same fact to filter out any false positive neurons that may represent other info (e.g. syntactic or lexical information) of one input fact. The idea is that different prompts corresponding to the same fact should share the same set of "true-positive" knowledge neurons since they express the same factual knowledge. They demonstrate that the activation of such knowledge neurons is positively correlated to the expression of their corresponding facts.

Following this, logically, after identifying where facts are located in the network, they can be updated or erased by changing the value of their knowledge neuron's contribution. The researchers performed an erasure case study, showing the model perplexity of the removed fact increased (so the LM is more unsure) but remained similar for unrelated facts. Although this is reassuring, they did not try to probe the network from the output side (for example, with a membership inference attack) so it is not certain that the information has been fully removed, and there is a privacy guarantee on this.

The Knowledge Neurons repository supports erase functionality for a given fact you wish to erase with the following procedure: (1) Curate a list of different expressions of this fact; (2) Identify knowledge neurons that appear most frequently among these facts; (3) Set these knowledge neurons to 0 i.e. zero vectors.

We carried out erasure experiments using the Knowledge Neurons repository with minimal changes for the following scenarios, which we thought were useful in a medical note setting: (1) Erase the link between a certain person and a disease e.g. Jane Smith has diabetes (to prevent privacy leakage of a certain patient with a certain disease); (2) Erase a certain Person (e.g. if someone practises their Right-to-be-Forgotten); (3) Erase a certain Disease (this could be useful in a non-privacy situation where this disease was so prevalent in the notes the dataset it is skewing the outputs of the LM). There are likely other healthcare scenarios we haven't considered, like updating a fact based on a correction to a patient diagnosis that was made since the notes were used for training. Experiments were performed using our Clinical-RoBERTa-None model, and examples from the 50% split of MIMIC-III were used for pre-training this model.

4.6 Techinical Implementation

4.6.1 Hardware

Pre-training, fine-tuning and membership inference were performed on a single machine hosted by Microsoft Azure with the following main specifications: 1 x NVIDIA Tesla T4 GPU, 4 x vCPUs from an AMD EPYC 7V12 processor, running a Linux Server.

4.6.2 Codebases

At the start of this project, there were several external codebases considered, and we have adapted and applied a wide range of them. We list key repositories that play a part or influence the work in Table 4.

Table 4: Summary of Codesbases Used in the Project.

Phase	Task	Codebases
RoBERTa Pre-training	Data Deduplication	https://github.com/ChenghaoMou/text-dedup
RoBERTa Pre-training	Pre-training	https://github.com/nhsx/ELM4PSIR
Nano-GPT Pre-training	Pre-training	https://github.com/karpathy/nanoGPT
RoBERTa Fine-tuning	Data Preparation	https://github.com/kexinhuang12345/clinicalBERT
Membership Inference Attacks	Calculating Sequence Scores	https://github.com/kanishkamisra/minicons
Post-Training Approaches	Knowledge Neurons	https://github.com/EleutherAI/knowledge-neurons

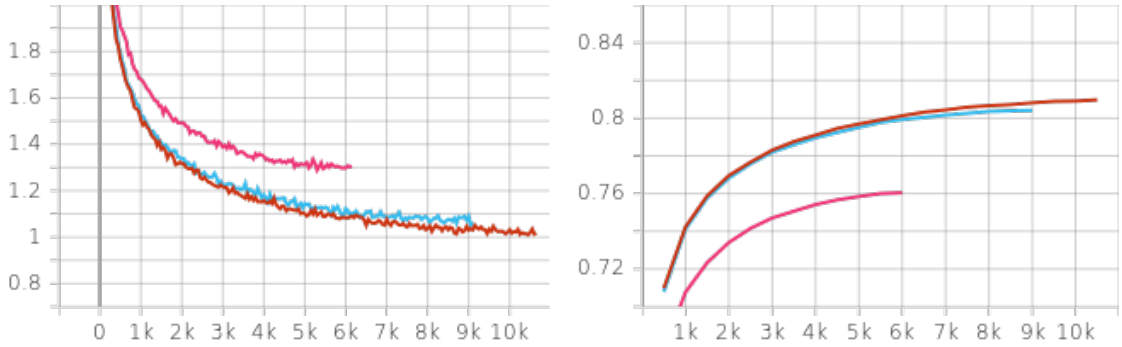
5 Results & Discussion

5.1 Semantic Deduplication

Unfortunately, we did not have time to take the Semantically Deduplicated Data through the full pipeline of experiments. Instead here, we share some insights from Semantic Clustering on MIMIC-III notes. Initially, using Top2Vec with distilBERT, we found that although some clusters were genuinely medically relevant, such as clustering sentences around respiratory conditions, others were not, such as clustering together all anagrams, many of which aren't semantically related. This is a result of the lack of medical acronym knowledge in distilBERT. We briefly explored if there were any methods to expand these acronyms in the text before passing this to Top2Vec. There is an interesting paper on training a T5 model to replace medical acronyms; however, they do not release the model for this [73]. They did release an acronym dictionary, which we explored using to replace acronyms with rules, however, without context, this was making many incorrect replacements, so we decided not to progress with this. We also explored the scispcacy entity linker [74], which uses UMLS to find acronym definitions, but found this to be very slow and so infeasible to use over the large number of notes we have. An interesting future project would be training a T5 model for decoding medical acronyms. We acknowledge that ontologies could also play a role in terms of synonyms, which we did not explore. For example, it would be interesting to explore if you replace medical synonyms, if sentence representations remain similar or change, and how this affects semantic deduplication. We found that BERTopic, which allowed us to use ClinicalBERT as the embedding model, was better able to cope with medical acronyms and produced more semantically appropriate clusters from a medical perspective.

5.2 Language Model Pre-training

Pre-training was continued from Roberta-base for one epoch, using our three datasets with different preprocessing. The results can be seen in Figure 4. We see the training loss goes down steeply and then plateaus as expected, and the evaluation accuracy increases over time before plateauing in all cases. As expected, the time for training Clinical-RoBERTa-NearDup is only slightly less than Clinical-RoBERTa-None (1̃7hours) as the NearDup technique only removes a small amount of data, and the Evaluation performance is very similar. Clinical-RoBERTa-ExactSubstr takes the least time to reach one epoch as the ExactSubstr Method removes a large amount of data (repeats). However, Clinical-RoBERTa-ExactSubstr does not reach as high Evaluation accuracy. This is unexpected as the original paper uses ExactSubstr to suggest models reach the same performance or can even show improved performance [28]. The difference we see here could be because of only training for one epoch and not allowing the model to see efficient data. Notably, we did not apply any deduplication to the evaluation sets, so this contains repeated phrases.



(a) Training Loss

(b) Evaluation Accuracy

Figure 4: Pre-Training Training Loss and Evaluation Accuracy Curves. Red= Clinical-RoBERTa-None, Blue = Clinical-RoBERTa-NearDup, Pink = Clinical-RoBERTa-ExactSubstr.

5.3 Downstream Tasks

As a sanity check for downstream task performance, the pre-trained model variations were fine-tuned on the MIMIC-III 30-day Readmission Prediction task [69]. The results are presented in Table 5. In the original ClinicalBERT paper, they achieved an AUC-ROC score of 0.714 ± 0.018 [69]. This result is not directly comparable to ours as they use 5-fold cross-validation and a scaling function over their scores to account for the number of notes per patient and note lengths. However, our results are in a similar ballpark to these ClinicalBERT, considering we only trained on 50% of MIMIC-III and for a single training epoch, unlike the original ClinicalBERT paper. Table 5 shows little difference

in performance on this task between our pre-trained model variations, suggesting the deduplication pre-processing does not affect downstream task performance. Interestingly, although the Clical-RoBERTa-ExactSubstr did not reach such high performance on the evaluation set in pre-training, this has not hindered the model's performance on the downstream tasks.

Table 5: 30-day Readmission Prediction Task Results.

Model	Deduplication	AUC-ROC
Clinical-RoBERTa	None	0.682 ± 0.002
	NearDup	0.679 ± 0.004
	ExactSubstr	0.681 ± 0.004
RoBERTa-base	None	0.679 ± 0.002

Results displayed here are evaluated on the Test Set and are an average of results over five fixed training runs with different shuffled training data.

As a second sanity check for downstream task performance, the pre-trained model variations were fine-tuned on the GLUE tasks [70]. The results are presented in Table 6. Here, we only report the results using fine-tuning and show the maximum score reached within five epochs. We only looked at a selection of the GLUE tasks which we deemed most relevant to our models. The tasks selected were: Standford Sentiment Treebank (SST-2) to determine the sentiment of the input text, The Corpus of Linguistic Acceptability (CoLA) with the objective of acceptability of the input sentences' grammar, and the Semantic TextualSimilarity benchmark (STS-B) to determine the similarity of two sentences with a score from 1-5. Table 6 shows our pre-trained models perform similarly across the board on tasks with RoBERTa base. This is in line with the 30-day Readmission results above. Again, although the Clinical-RoBERTa-ExactSubstr did not reach such high performance on the evaluation set in pre-training, this has not hindered the model's performance on the downstream tasks, where it even performs slightly better than RoBERTa-base on the CoLA task.

Table 6: Evaliation Metrics for Models Fine Tuned on the GLUE Tasks: SST-2, STS-B, and CoLA.

Model	Deduplication	SST-2 -Accuracy	CoLA - Matts. corr.	STS-B -Spearmans r
Clinical-RoBERTa	None	0.94	0.60	0.90
	NearDup	0.94	0.60	0.91
	ExactSubstr	0.94	0.62	0.90
RoBERTa-base	None	0.95	0.61	0.90

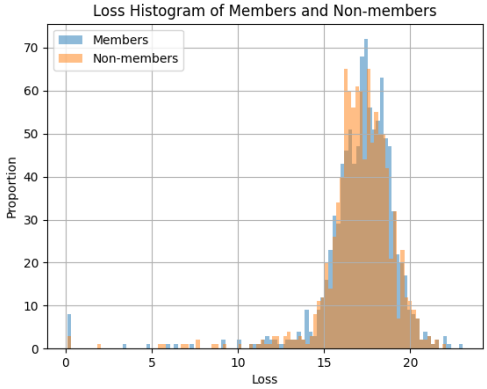
Results displaying maximum performance in 5 epochs.

5.4 MIAs on Models Pre-trained on data with different Pre-processing

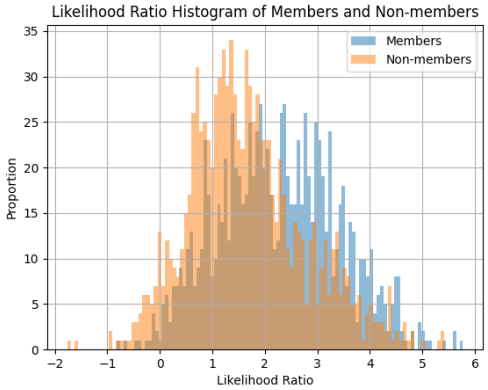
Loss-based and Likelihood-ratio (LR)-based Membership Inference Attacks (MIAs) were conducted on our pre-trained models.

Figure 5 shows the output distributions of loss-based compared with LR-based MIAs. For both the non-member (MIMIC-III) and external (i2b2) datasets, the loss-based attack shows highly overlapping distributions with the member dataset, demonstrating it does not distinguish well between members and texts the model has never seen before. Constrasitvely, LR-based MIAs are consistently better, showing more distinguishable distributions, particularly pronounced for the externals compared to members. Table 7 shows the threshold-independent AUCs, which confirm LR-based attacks are stronger compared with their loss-based counterparts. This is likely as employing a reference model allows for better detection of specific sequences from the training data, whilst reducing the signal of general sequences which the model will have seen before but are not of interest.

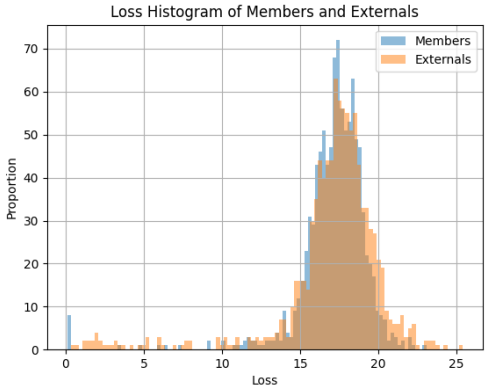
Figure 6 shows the output distributions of LR-MIAs against our pre-trained model using two reference models-RoBERTa-base and RoBERTa-base-PM-Voc. We did not observe much difference between reference model distributions for non-members or externals. Table 7 shows the threshold-independent AUCs. Please note that as the loss-based attack does not employ a reference model and uses only the member distribution loss, the scores for the two reference model rows are the same. It can be observed that RoBERTa-base-PM-Voc offers a slight improvement over RoBERTa-base. This is to be expected as the distribution of data RoBERTa-base-PM-Voc is trained on (PubMed) is more similar in the domain to MIMIC-III and so likely it can better counter "common" phrases from this domain and prevent these being included as False Positives.



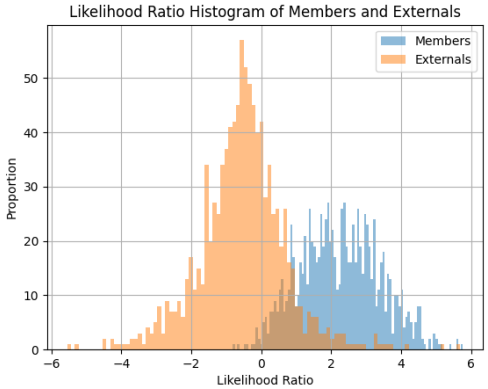
(a) Loss, Non-Members



(b) LR, Non-members



(c) Loss, Externals

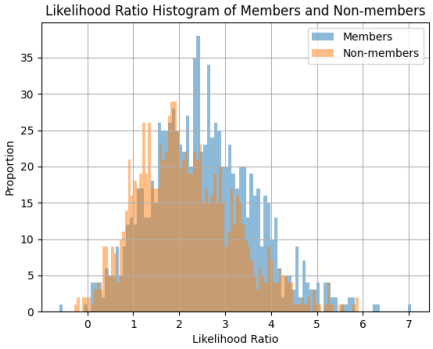


(d) LR, Externals

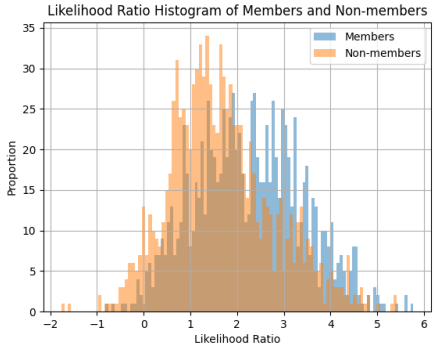
Figure 5: MIA Distributions between Loss and LR-based Attacks for Non-Member and External Datasets.Target Model was Clinical-RoBERTa-None and Reference Model was RoBERTa-base-PM-Voc.

Figure 7 shows the plots of LR-based distributions for the models trained on data with different preprocessing. These are distributions are quite similar between preprocessing techniques for both non-members and externals. Table 7, showing the threshold-independent AUCs, confirms that little difference is seen for either non-members or externals. Previous studies using GPT models and public domain data, have shown deduplication techniques reduce memorization significantly [27, 6]. However, no such studies have been performed on Masked Language Models, so there could be a difference stemming from this. As the models used were generative, previous studies could also use Extraction Attacks [27, 6], which are slightly stronger than MIAs. Another possibility is that we only trained our models for one epoch. Perhaps training for more epochs or overfitting the models would tease out these differences. Interestingly, we see more notable differences between AUCs of external loss-based attacks (Table 7), with no preprocessing being the most vulnerable and the ExactSubstr pre-processing being the least vulnerable (lowest AUC of 0.65).

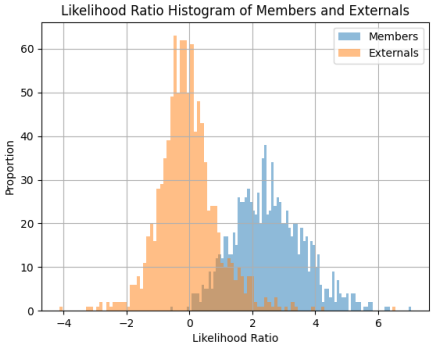
As a thought experiment, we also ran MIAs using only our pre-trained models as both target and reference models. In all cases, we used the model trained with a larger subset of data as the reference model. Table 8 shows the threshold-independent AUCs for the likelihood ratio attacks. Please note that the loss-based attack does not employ a reference model and uses only the member distribution loss; the scores are the same for each target model as in Table 7 above and so have not been included.



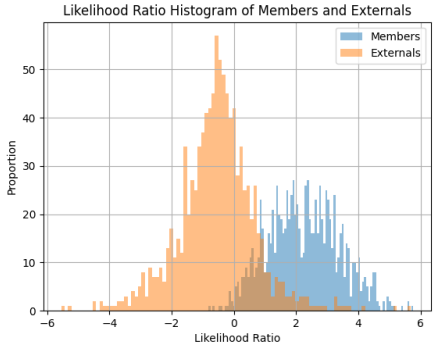
(a) LR, Non-members, RoBERTa-base



(b) LR, Non-members, RoBERTa-base-PM-Voc



(c) LR, Externals, RoBERTa-base



(d) LR, Externals, RoBERTa-base-PM-Voc

Figure 6: MIA Distributions between Different Reference Models.

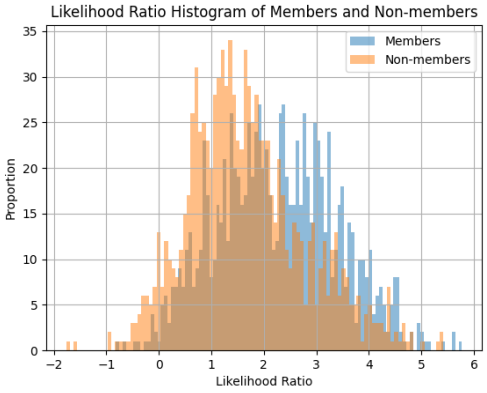
Table 7: Threshold-Independent AUC showing Privacy Risk with MIAs.

Target Model	**Reference Model**	**Non-Members**		**Externals**	
		Loss	LR	Loss	LR
Clinical-RoBERTa-None	RoBERTa-base	0.64	0.71	0.80	0.99
	RoBERTa-base-PM-Voc	0.64	0.76	0.80	1.0
Clinical-RoBERTa-NearDup	RoBERTa-base	0.61	0.74	0.70	0.99
	RoBERTa-base-PM-Voc	0.61	0.77	0.70	1.0
Clinical-RoBERTa-ExactSubstr	RoBERTa-base	0.63	0.72	0.65	0.99
	RoBERTa-base-PM-Voc	0.63	0.75	0.65	1.0

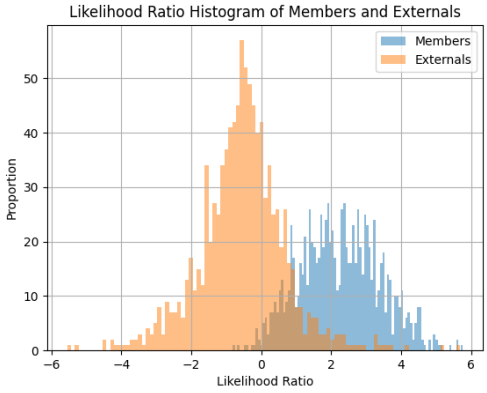
Please note that as the loss-based attack does not employ a reference model and uses only the member distribution loss, the scores for the two reference model rows are the same.

Table 8: Threshold-Independent AUC showing Privacy Risk between our Pre-trained Models.

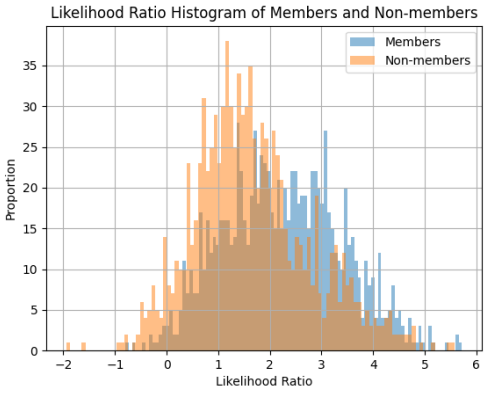
Target Model	**Reference Model**	**Non-Member LR**	**External LR**
Clinical-RoBERTa-NearDup	Clinical-RoBERTa-None	0.60	0.68
Clinical-RoBERTa-ExactSubstr	Clinical-RoBERTa-None	0.65	0.87
	Clinical-RoBERTa-NearDup	0.67	0.87



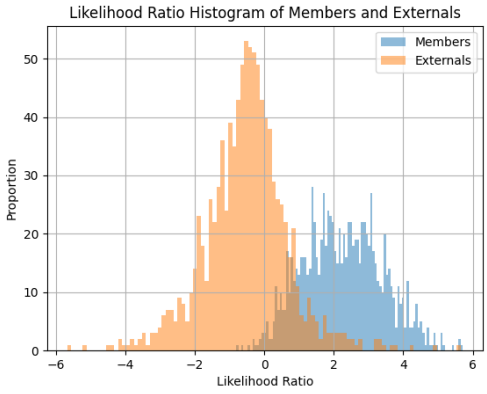
(a) LR, None, Non-members



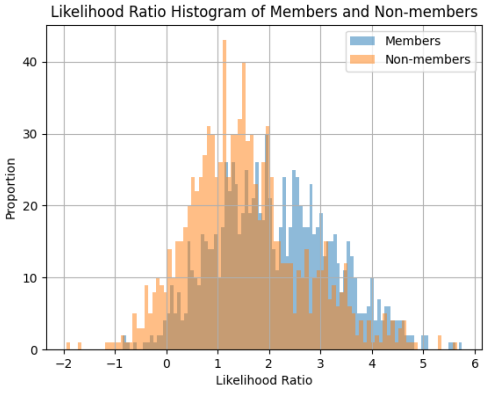
(b) LR, None, Externals



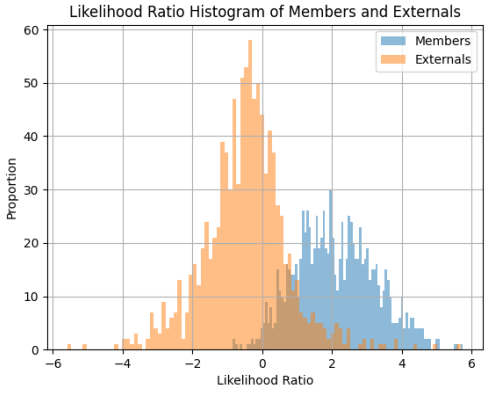
(c) LR, NearDup, Non-members



(d) LR, NearDup, Externals



(e) LR, ExactSubstr, Non-members



(f) LR, ExactSubstr, Externals

Figure 7: LR-MIA Distributions between Models Trained on Data with different Deduplication Techniques.

5.5 Machine Unlearning and Editing

We had some success with the method, changing the argmax completion of an input phrase relating a patient to a disease. For example, given the phrase “Jane Smith has intra-abdominal collateral vessels consistent with underlying portal hypertension", using the Knowledge Neurons Method, we identified a small set of neurons associated with the relational fact and set these to zero to effectively erase this fact. Upon passing this phrase to the model with the disease masked, the argmax prediction is for a different disease or just the “disease" itself. However, we found some limitations when applying the method: (1) Only one word can be erased at a time in a phrase; this is not ideal in a medical note setting, where diseases can be many words; (2) You need to curate a list of alternative ways of expressing your fact, this is difficult to do, and the success of the method relies on you doing this comprehensively. For medical facts relating person and disease, it was particularly difficult to do this. (3) the model often did not have the highest probability (the argmax completion) for the true disease or patient upon passing in the original fact before erasure. This is likely because, in MIMIC-III, most patients have many multi-morbidities, which often appear as lists within the notes, making it difficult for the model to learn how to autocomplete a phrase of “patient X has [MASK]". This is a serious limitation as it relies on the model knowing a certain fact in that format of completing it to carry out and test the method, but the model could still be vulnerable to MIAs or other privacy attacks on these phrases even if the model doesn’t predict the MASK correctly in this setting.

These findings line with Lehman et al. [43], who find they can only extract 7% of patient-disease pairs correctly from ClinicalBERT when they probe the model with phrases like “Jane Smith has [MASK] disease" and use the argmax predictions to fill in the blank. They found that the model’s predictions mostly align with disease frequencies in the dataset, and so could not mount a strong attack with this method. However, MIAs on ClinicalBERT have achieved a high AUC of 90% [36]. This demonstrates that even if a LM cannot correctly predict a masked word in an input sequence, it does not mean that information about that “fact" is not stored in the network and cannot be extracted or inferred with strong privacy attacks. This thinking calls the Knowledge Neurons Method into question as a privacy-preserving method as it is not clear that changing the model prediction probabilities over a certain relational fact is akin to removing the fact from the network together and, therefore, protecting that fact. So perhaps factual editing is a little bit difficult regarding privacy in this context, particularly where patients have many diseases. The method is more suited to updating facts in medical data based on new research or new definitions.

Unfortunately, we ran out of time to apply MIAs to the facts we had “erased" from our model using Knowledge Neurons. However, anyway, it as hard to gather more than a handful of facts to which we could apply this technique, as the facts did not get predicted correctly as described above. Looking to the future, methods which allow the unlearning or editing of specific training examples should be given primary focus from a privacy perspective. Methods allowing model unlearning of wider concepts across the dataset also hold potential. All editing and unlearning-based methods should be evaluated with privacy attacks to understand the method’s effectiveness from a privacy perspective (rather than just a prediction/auto-complete perspective).

6 Conclusion

Throughout this work, we explored the Privacy Risks and Mitigations for Healthcare LMs. We sought to understand more about the Privacy-Risk Landscape for Healthcare LMs and conduct a practical investigation on some existing attack and defensive methods. Initially, we conducted a thorough literature search to understand the privacy risk landscape. The outputs of this are available in Appendix A.

In our first applied work package, we explored data deduplication before model training as a mitigation to reduce memorization and evaluated the approach with Membership Inference Attacks (MIAs). We showed that RoBERTa models trained on patient notes are highly vulnerable to Likelihood Ratio MIAs, even when only trained for a single epoch. We investigated data deduplication as a mitigation strategy but found that these models were just as vulnerable to MIAs. Further investigation of models trained for multiple epochs is needed to confirm these results. In the future, semantic deduplication could be a promising avenue for medical notes.

In our second applied work package, we explored editing/unlearning approaches for healthcare LMs. Unlearning in LMs is poised to become increasingly relevant, especially in light of the growing awareness surrounding training data leakage and the Right to be Forgotten. We found that many repositories were not adapted for all LM types, and some are still not mature enough to be easy to use as packages. Exploring a Locate-then-Edit approach to Knowledge Neurosn, we found this was not well suited to the erasure of information we needed in medical notes. Our findings suggest that the focus from a privacy perspective on these methods should be on those which allow the erasure of specific training data instances instead of relational facts.

7 Future Work

Investigating further training. In this study, models were limited to a single training epoch, a restriction imposed by time constraints. Yet, in practical applications, models are typically subjected to multiple training epochs to optimize their performance. This practice holds significance concerning privacy risks, as models tend to memorize data points encountered repeatedly [6]. Therefore, it becomes imperative to understand the interplay between training duration, privacy risk, and the effectiveness of mitigation strategies.

Expanding the work to other LMs. This work primarily explored privacy in the Masked Language Model RoBERTa. Expanding this work to the wealth of other Encoder and Encoder-Decoder models like GPT and T5 is essential. The relevance of such an extension is underscored by the growing adoption of these model types and existing research, which has already shed light on the privacy risks associated with these models [23, 37].

Expanding the work on Unlearning & Editing. Although within the scope of this study, the field of Machine Unlearning/Editing was in its infancy, it is gaining momentum. As this field matures, it becomes crucial to focus on comparing the efficacy of different methods. Furthermore, it is important to explore the effect of removing the influence of a set of data points. This could potentially facilitate easier data extraction of other training data. Therefore, a holistic examination of the effectiveness, privacy implications, and broader impacts of Machine Unlearning/Editing methods on healthcare LMs is essential to inform the development of robust and privacy-conscious LMs in the NHS.

Investigating Mitigation Strategies on Fine-tuned Models. This work exclusively explored privacy mitigations on LMs with continued pre-training (across all parameters). Due to the common practice of freezing parameters and tuning the last layer of a LM on a private dataset, it is important to investigate the privacy risk when fine-tuning. Consequently, there is a critical need to explore the privacy implications associated with the fine-tuning process. Pre-processing mitigation strategies should be applied to fine-tuning data. Additionally, unlearning/editing strategies for data included in the fine-tuning dataset of a fine-tuned LM should be tested.

Expanding the work on Semantic Deduplication. Semantic Deduplication holds significant promise in addressing memorization issues and mitigating the risks associated with training data leakage in Language Models (LMs). Expanding the current work would involve pre-training a model utilizing semantically deduplicated datasets and subjecting the model to MIAs. Given the low success of other deduplication methods in protecting the model from LR-MIAs, it would be interesting to see if semantic deduplication can offer enhanced privacy protection.

Retrieval Augmented LMs. It has been demonstrated that a LM's ability to answer a fact-based question relates to how many documents associated with that question were seen during pre-training [75]. LMs are prone to "hallucination" in cases where the LM has not seen many documents in pre-training. External Knowledge Bases can enhance LMs by providing external knowledge for inference and interoperability, known as retrieval augmentation. This could have

particular utility across Healthcare. However, one study investigating privacy in this scenario found the knowledge base enhanced LM was more likely to leak private information [76]. This finding underscores the critical need for a nuanced understanding of the privacy risks of Retrieval Augmented LMs, especially in healthcare scenarios where sensitive data is involved.

MultiModal Models As real-world data exist in different modalities, this has sparked the emergent research into the Multimodal Large Language Model (MLLM), referring to LM-based models which can take in and reason over multimodal information [77]. In healthcare, where patient data frequently manifests in diverse forms such as medical images, time series, medical notes, and structured data, the utility of MLLMs becomes particularly evident. However, as demonstrated in recent research [78], MLLMs are susceptible to membership inference attacks, a vulnerability that results in information leakage from one input modality through another output modality. It becomes imperative to delve deeper into understanding this risk, especially within healthcare scenarios.

Privacy-Explainability-Fairness. Explainability often involves generating explanations or counterfactuals alongside the decisions made by the LM. However, integrating explanations into the output of LMs can introduce vulnerabilities related to training data leakage and privacy attacks. Additionally, efforts to enhance privacy, such as employing Differentially Private (DP) training techniques, can inadvertently impact fairness, particularly in datasets lacking diversity. Text that is underrepresented in training data receives larger updates in training and thus is more affected by clipping and noising in DP-SGD. Intuitively, DP-SGD training amplifies a model's "bias" towards the most popular distribution elements being learned [79]. Consequently, the trade-off between privacy, explainability, and fairness becomes a complex challenge. Investigating this privacy-explainability-fairness trade-off is crucial for developing robust and ethically sound machine learning models, especially in healthcare, where all three elements are paramount.

Please see Appendix A for further information and resources on the areas mentioned above.

References

[1] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Anthony B Costa, Mona G Flores, et al. A large language model for electronic health records. *npj Digital Medicine*, 5(1):194, 2022.

[2] Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful zero-shot predictors of non-coding variant effects. *bioRxiv*, pages 2022–08, 2022.

[3] Hong-Liang Li, Yi-He Pang, and Bin Liu. Bioseq-blm: a platform for analyzing dna, rna and protein sequences based on biological language models. *Nucleic acids research*, 49(22):e129–e129, 2021.

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

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

[6] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[7] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

[8] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

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

[13] Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. Extracting multiple-relations in one-pass with pre-trained transformers. *arXiv preprint arXiv:1902.01030*, 2019.

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

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[18] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.

[19] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[20] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*, 2019.

[21] Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, Zumrut Muftuoglu, Sahib Singh, and Fatemehsadat Mireshghallah. Benchmarking differential privacy and federated learning for bert models. *arXiv preprint arXiv:2106.13973*, 2021.

[22] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[23] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

[24] Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*, pages 273–281. Springer, 2020.

[25] Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

[26] R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *arXiv preprint arXiv:2111.09509*, 2021.

[27] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[28] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

[29] Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*, 2022.

[30] Victoria Smith, Ali Shahin Shamsabadi, Carolyn Ashurst, and Adrian Weller. Identifying and mitigating privacy risks stemming from language models: A survey. *arXiv preprint arXiv:2310.01424*, 2023.

[31] Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

[32] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[33] Sorami Hisamoto, Matt Post, and Kevin Duh. 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, 2020.

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

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

[36] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

[37] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*, 2022.

[38] Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. Masked language model scoring. *arXiv preprint arXiv:1910.14659*, 2019.

[39] Carina Kauf and Anna Ivanova. A better way to do masked language model scoring. *arXiv preprint arXiv:2305.10588*, 2023.

[40] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

[41] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[42] Mohamed Abdalla, Moustafa Abdalla, Graeme Hirst, and Frank Rudzicz. Exploring the privacy-preserving properties of word embeddings: algorithmic validation study. *Journal of medical Internet research*, 22(7):e18055, 2020.

[43] Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. Does bert pretrained on clinical notes reveal sensitive data? *arXiv preprint arXiv:2104.07762*, 2021.

[44] Huseyin A Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. Training data leakage analysis in language models. *arXiv preprint arXiv:2101.05405*, 2021.

[45] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019.

[46] Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*, 2020.

[47] Chen Chen, Xuanli He, Lingjuan Lyu, and Fangzhao Wu. Killing one bird with two stones: Model extraction and attribute inference attacks against bert-based apis. *arXiv preprint arXiv:2105.10909*, 2021.

[48] Tuomas Aura, Thomas A Kuhn, and Michael Roe. Scanning electronic documents for personally identifiable information. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 41–50, 2006.

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

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

[51] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013.

[52] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[53] Shlomo Hoory, Amir Feder, Avichai Tendler, Sofia Erell, Alon Peled-Cohen, Itay Laish, Hootan Nakhost, Uri Stemmer, Ayelet Benjamini, Avinatan Hassidim, et al. Learning and evaluating a differentially private pre-trained language model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1178–1189, 2021.

[54] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[55] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

[56] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[57] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

[58] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[59] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*, 2023.

[60] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

[61] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

[62] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[63] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

[64] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[65] Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of biomedical informatics*, 58:S20–S29, 2015.

[66] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

[67] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

[68]

[69] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

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

[71] Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*, 2023.

[72] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[73] Alvin Rajkomar, Eric Loreaux, Yuchen Liu, Jonas Kemp, Benny Li, Ming-Jun Chen, Yi Zhang, Afroz Mohiuddin, and Juraj Gottweis. Deciphering clinical abbreviations with a privacy protecting machine learning system. *Nature Communications*, 13(1):7456, 2022.

[74] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics.

[75] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. *arXiv preprint arXiv:2211.08411*, 2022.

[76] Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. Privacy implications of retrieval-based language models. 5 2023.

[77] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[78] Pingyi Hu, Zihan Wang, Ruoxi Sun, Hu Wang, and Minhui Xue. M4i: Multi-modal models membership inference. *Advances in Neural Information Processing Systems*, 35:1867–1882, 2022.

[79] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

[80] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[81] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[82] Zhe Liu, Xuedong Zhang, and Fuchun Peng. Mitigating unintended memorization in language models via alternating teaching. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[83] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*, 2023.

[84] Martin Pawelczyk, Himabindu Lakkaraju, and Seth Neel. On the privacy risks of algorithmic recourse. 11 2022.

[85] Saifullah Saifullah, Dominique Mercier, Adriano Lucieri, Andreas Dengel, and Sheraz Ahmed. Privacy meets explainability: A comprehensive impact benchmark. 11 2022.

[86] Cleo Matzken, Steffen Eger, and Ivan Habernal. Trade-offs between fairness and privacy in language modeling. 2023.

[87] Khang Tran, Ferdinando Fioretto, Issa Khalil, My T Thai, and NhatHai Phan. Fairdp: Certified fairness with differential privacy. *arXiv preprint arXiv:2305.16474*, 2023.

[88] John X Morris, Justin T Chiu, Ramin Zabih, and Alexander M Rush. Unsupervised text deidentification. *arXiv preprint arXiv:2210.11528*, 2022.

[89] My H Dinh and Ferdinando Fioretto. Context-aware differential privacy for language modeling. *arXiv preprint arXiv:2301.12288*, 2023.

[90] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[91] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

[92] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

[93] Zhichao Yang, Shufan Wang, Bhanu Pratap Singh Rawat, Avijit Mitra, and Hong Yu. Knowledge injected prompt based fine-tuning for multi-label few-shot icd coding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2022, page 1767. NIH Public Access, 2022.

A Appendix

Before deciding on the project direction, a literature search was performed across a wide range of topics in LM Privacy. This Section includes Notes from that exploration across a wide range of topics. Some of these research directions were not mature enough for us to pursue this topic, but could be important areas to watch going forward.

A.1 Investigating Machine Unlearning in Healthcare LMs

NHS/Healthcare Scenario where this is Relevant

Specific Users want their Healthcare Data Removed from a trained Model, e.g. The National Data Opt-Out. From a legal perspective, there is a need to have a methodology to remove the influence of user data from trained models easily. It is also relevant to remove data poisoning from a model.

Background

Large Language Models (LLM) are increasingly being trained on vast, minimally curated datasets which may contain a user's data. To protect a user's privacy, the influence of a specific data point used to train the model can be requested to be erased. Retraining the LLM from scratch each time this is requested is not a plausible approach due to the cost and resources involved. Machine Unlearning refers to an emerging set of techniques to remove the influence of specific training examples on the trained model- this has been termed as the forget set. The overarching aim is to preserve model performance and generalization whilst removing the influence of specific training examples to preserve the privacy of these examples (ensure they cannot be "leaked" from the LLM). To avoid retraining LLMs from scratch, unlearning methods adjust the fully trained LLM to remove the influence of the forget set e.g., by adding noise to the model weights.

Notably, the evaluation of forgetting algorithms in the literature has been inconsistent using a mixture of distance to the fully trained model, classification accuracy on samples to unlearn, and error rate of membership inference attacks. The NeurIPS first challenge on machine unlearning suggests using Likelihood ratio attacks [80] (a form of MIA), to evaluate machine unlearning, deeming it successful if the adversary cannot infer the presence of the forget set in the training examples. They also propose using statistical tests to quantify the difference in the distribution of unlearned models compared to the distribution of models retrained from scratch. For an ideal unlearning algorithm, these two will be indistinguishable.

Machine Unlearning has been attempted in LLMs in the general domain, so-called "Knowledge Unlearning" [55]. The authors perform gradient ascent, which can be thought of as maximising the loss function, on target token sequences, which they want GPT-Neo to "forget". They found their method was sufficient to protect their target sequences from extraction attacks with negligible degradation in performance (and, in some cases, improvements of up to 10%!).

Resources

1. Machine Unlearning NeurIPs Challenge
2. On the Necessity of Auditable Algorithmic Definitions for Machine Unlearning [81]
3. Knowledge Unlearning for Mitigating Privacy Risks in Language Models [55]
4. Concept Erasure with Least-squares
5. Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy
6. When Machine Unlearning Jeopardizes Privacy
7. To Be Forgotten or To Be Fair: Unveiling Fairness Implications of Machine Unlearning Methods
8. Knowledge Neurons in Pretrained Transformers
9. Privacy Adhering Machine Un-learning in NLP
10. Gradient Ascent on target token sequences (minimize negative log-likelihood) `https://arxiv.org/pdf/2210.01504.pdf`
11. SISA - The framework provides a strategic way to limit the influence of a data point in the training procedure. The approach trains models in isolation on disjoint shards created by partitioning the training data, adapted to NLP by storing only task-specific layers or using adapters and storing only adapter weights instead of entire model checkpoints. (There also DeepObliviate, which is similar to this technique but has not been adapted to LMs)`https://arxiv.org/abs/2105.06209`.

12. Combine with pruning methods -unlearning on a sparse model can lead to a smaller unlearning error `https://arxiv.org/pdf/2305.06360.pdf`
13. Use a teacher network to guide fine-tuning of a student network on a small subset of clean training data. `https://arxiv.org/pdf/2305.06360.pdf`
14. To unlearn concepts - concept scrubbing from LEACE `https://arxiv.org/pdf/2306.03819.pdf`
15. Packages for Privacy Attacks: Text Attack GitHub
16. Papers with Code: Summary of MIA Papers with Code
17. Machine Unlearning Packages: Summary of MU papers & methods, Knowledge Unlearning Paper
18. Data Cartography - which points are hardest to unlearn?

A.2 Investigating Memorization in Healthcare LMs

NHS/Healthcare Scenario where this is Relevant Any LMs trained on private patient or staff data will be released to a wider audience than those with direct access to the training data.

Background

It has been demonstrated that LMs memorize aspects of their training data, which can be extracted verbatim when LMs are prompted, referred to as training data leakage. Training data memorization can violate the privacy assumptions under which datasets were originally collected and make disparate information more readily searchable. As LMs have grown in size, the increasing over-parameterization of models increasingly enables this memorisation of parts of the training data

Memorization has been shown to scale with (1) Model size (number of parameters), (2) duplicated sequences in the dataset and (3) the number of tokens in the prompt fed to the LM [6, 27].

Notably, LMs have also been shown to emit paraphrased content (through malicious or honest interactions with the LM)[66]. Preventing verbatim memorization does not present training data leakage in LMs. At each generation step, they check whether the model's chosen next token would create an n-gram found in the training set. If it does, an alternative next token is selected. When prevented from generating exact n-grams from the training set, LMs can " cheat" the filter by producing close paraphrases–for example, inserting spelling errors, adjusting punctuation or whitespace, or using synonyms. These changes lead to generated text a human would perceive as nearly identical, even if it is not verbatim memorization. They, therefore, argue that a broader definition of memorization is necessary when reasoning about training set memorization in language models.

Data deduplication effectively safeguards against training data leakage from large LMs, rendering model inversion attacks largely ineffective [27, 28]. However, Data Deduplication reduces memorization on average and therefore cannot guarantee to prevent memorization of a specific training example.

Alternate teaching has also been used to mitigate unintended memorization [82]. Multiple teachers are trained on disjoint training sets (from different users) whose privacy one wishes to protect. Teachers' predictions supervise the training of a student model in an alternating manner at each time step. At each time step, they only leverage the prediction output from one teacher to disconnect consecutive words in private sequences but generally have no issues in learning common and non-sensitive sequences that are present in the majority of teachers.

Attempts to predict which samples will be memorized from checkpoints of full model [83] have also been explored. Training much smaller models was not a good indicator, so needed checkpoints that were 50% of the size of the full model.

Resources

1. Quantifying Memorization Across Neural Language Models
2. Deduplicating Training Data Mitigates Privacy Risks in Language Models
3. Deduplicating Training Data Makes Language Models Better
4. Preventing Verbatim Memorization in Language Models Gives a False Sense of Privacy
5. Large Language Models Struggle to Learn Long-Tail Knowledge
6. MITIGATING UNINTENDED MEMORIZATION IN LANGUAGE MODELS VIA ALTERNATING TEACHING

7. Textbooks Are All You Need -the idea of higher quality data breaking existing scaling laws for performance in LLMs.
8. Data Cartography
9. A framework for transparent and accessible large-scale language model training
10. `https://github.com/google-research/deduplicate-text-datasets`
11. `https://github.com/nkandpa2/long_tail_knowledge`
12. Data Cartography

A.3 Investigating Privacy of Multimodal Healthcare Models

Proposed Work Plan

NHS/Healthcare Scenario where this is Relevant Generating descriptions for Medical Images or parts of a video (e.g. during surgery). Reasoning over medical images or videos to make decisions (e.g. patient triaging, patient care plans, etc.). Medical report generation or disease identification.

Background With the development of machine learning techniques, the attention of researchers has been moved from single-modal learning to multi-modal learning, as real-world data exist in different modalities. There have also been huge recent advances in LLMs. This has sparked the emergent research into the Multimodal Large Language Model (MLLM), referring to LLM-based models which are able to take in and reason over multimodal information. For example, LLMs can write captions or stories based on images and even interpret memes [77].

As it would be computationally costly to train a multimodal model end-to-end, the challenge comes with bridging the gap between LLMs and inputs from other modalities. Instead, a learnable interface between the LLM and the encoder of the other data type (e.g. visual encoder) is used, e.g. using a linear layer or shallow MLP to embed image features. Alternatively, other modalities can be translated into language using expert models before inputting into the LLM, e.g. using an image captioning model. However, notably, this causes information loss from the image.

Prompting can incorporate multimodal data into LLM reasoning, for example, across video footage, asking the LLM to "think frame by frame" or ask what happened between two specific keyframes. In this way, the models can learn to leverage embedded knowledge and reasoning ability without explicit guidance.

Multimodal models cause privacy concerns, as they often carry more information than unimodal models. They are also often applied in situations with highly sensitive data, such as medical report generation or disease identification. [78] demonstrate that multimodal models are vulnerable to leaking training data through the lens of membership inference attacks. They employ two versions of this membership inference attack: (1) metric-based (MB), adopting similarity metrics while attacking to infer target data membership, and (2) feature-based (FB), uses a pre-trained shadow multi-modal feature extractor to achieve the purpose of data inference attack by comparing the similarities from extracted input and output features. Their attacks achieve strong performance (72-95% success rate). They use an encoder-decoder architecture with CNN layers to encode the image and LSTM layer for text generation.

Resources/Papers

1. A Survey on Multimodal Large Language Models
2. `https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models`
3. Integrated multimodal artificial intelligence framework for healthcare applications
4. Generating Images with Multimodal Language Models
5. MIMIC-IT: Multi-Modal In-Context Instruction Tuning
6. `https://github.com/HenryHZY/Awesome-Multimodal-LLM`
7. X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages
8. Privacy Distillation: Reducing Re-identification Risk of Multimodal Diffusion Models
9. On the Privacy Risks of Model Explanations
10. M4I: Multi-modal Models Membership Inference
11. A multimodal differential privacy framework based on fusion representation learning

12. There are a number of multimodal datasets here. However, none of these are medical-specific. There is also MedICaT, which has medical images from PubMed papers and associated captions. Also, paired images and radiological reports from the MIMICCXR 2.0.0 database.
13. `https://github.com/MultimodalMI/Multimodal-membership-inference.git`

A.4 Investigating the Privacy-Explainability Interaction in Healthcare LMs

NHS/Healthcare Scenario where this is Relevant On any LM where you wish to understand further the output/decision made.

Background As LMs become more powerful and are deployed in more real-world contexts, understanding their behaviour is critical. Researchers are interested in the behaviour of these models under domain shift and adversarial settings, but also their tendencies to behave according to social biases or shallow heuristics. For any new LM, we want to know cases where it performs poorly, why it makes a particular prediction, or whether a model will behave consistently under varying inputs.

Explainability often involves getting LMs to output explanations alongside decisions made (or counterfactuals). These can be subject to training data leakage and privacy attacks. Privacy concerns of counterfactuals have been explored through counterfactual-based distance attacks [84]. Here, the distance between the instance and its corresponding counterfactual captures information about whether the instance was used to train a model. The effect of private training on generated explanations has also been studied using images and time-series data [85].

Resources

1. Privacy Meets Explainability: A Comprehensive Impact Benchmark
2. ON THE PRIVACY RISKS OF ALGORITHMIC RECOURSE
3. `https://ai.googleblog.com/2020/11/the-language-interpretability-tool-lit.html` & The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models - e.g. They use a T5 model to summarize the text. For an example of interest, quickly find similar examples from the training set using an approximate nearest-neighbours index.
4. Language Model Behavior: A Comprehensive Survey
5. Explainability Techniques for Chemical Language Models
6. `https://github.com/pair-code/lit` - can explore explainability and adversarial attacks.

A.5 Investigating the Privacy-Utility-Fairness Trade-Off in Healthcare LMs

NHS/Healthcare Scenario where this is Relevant Text Classification Models where there are minority training data groups. Maybe in general, medical text data does not hold biases in Language (but could be notable for performance on minority groups where fairness is less of a concern).

Background Studies have reported that when models are trained on data with long-tailed distributions, it is challenging to develop a private learning algorithm that has high accuracy for minority groups. In this way, Differentially Private (DP) training can negatively impact fairness when data lacks diversity. The language that is underrepresented in training data receives larger updates in training and thus is more affected by clipping and noising in DP-SGD.

Intuitively, DP-SGD amplifies a model's "bias" towards the most popular elements of the distribution being learned. This effect was demonstrated in BiLSTM with pre-trained GloVe embeddings [79]. The authors show that the reduction in accuracy of tweet classification incurred by the DP model disproportionately impacts underrepresented subgroups, as well as subgroups with relatively complex data when using the dataset on Twitter posts from a corpus of African-American English. Conversely, [86] demonstrated DP training on GPT-2 did not harm fairness using the Bec-Pro gender bias in professions dataset, as this dataset has no quantitative minority group.

There have not been any works looking at countering this effect in LMs. However, FairDP [87], applies techniques to counter the negative effects of DP on minority groups in MLPs using tabular data. They train group-specific models and progressively aggregate group models to construct a general model, controlling noise injection per group.

Resources

1. Differential Privacy Has Disparate Impact on Model Accuracy in LMs

2. Trade-offs between fairness and privacy in LMs
3. FairDP
4. Some background on fairness in LLMs: 1 &, 2 & 3.

A.6 Investigating Privacy Preserving Training in Healthcare LMs

NHS/Healthcare Scenario where this is Relevant Training any model with a wider audience than its training data and where you want to prevent memorization of individual training data points with firm guarantees.

Background An algorithm is differentially private (DP) if its outputs are statistically indistinguishable on neighbouring datasets that differ by only one record. DP stochastic gradient descent (DP-SGD) is the primary approach to training ML models with DP. DP-SGD makes two modifications to vanilla SGD in each training iteration: (1) per-example gradients are clipped to a fixed norm, C, to bound the impact of individual training examples on model updates (2) calibrated Gaussian noise is added proportional to C to the aggregated clipped gradients. The more noise is added, relative to the clipping norm, the more strict the upper bound on the privacy loss that can be guaranteed.

DP-SGD can cause a significant reduction in accuracy due to applying gradient clipping and noising at the granularity of individual training examples. In addition, computing per-example gradients in DP-SGD incurs significant memory and computational overhead. This makes DP-SGD challenging for transformer-based LMs, which typically have hundreds of millions of parameters.

Resources

1. Just Fine-tune Twice: Selective Differential Privacy for Large Language Models Selective Differential Privacy (SDP) to protect only the sensitive tokens defined by a policy function. First, fine-tunes the model with redacted in-domain data, and then fine-tune it again with the original in-domain data using a private training mechanism. Adaptive Differential Privacy for Language Modeling estimates the probability that a linguistic item contains privacy based on a LM - assume private sequences do not occur frequently (so the probability that a linguistic item contains privacy information is inversely proportional to the frequency of the linguistic item occurring in the dataset propose a new Adam algorithm that adjusts the degree of differential privacy noise injected to the language model according to the estimated privacy probabilities.
2. Work to improve the efficiency of DP-SGD like pre-training and fine-tuning of LLMs: Large-Scale Differentially Private BERT, Differentially Private Fine-tuning of Language Models, Large Language Models Can Be Strong Differentially Private Learners, Large Scale Private Learning via Low-rank Reparametrization and for model compression. Differentially Private Model Compression.
3. Training Large-Vocabulary Neural Language Models by Private Federated Learning for Resource-Constrained Devices. Train LLM on compute-constrained devices while preserving privacy using FL and DP. DP-noise introduced to the model increases as the model size grows, which often prevents convergence propose Partial Embedding Updates (PEU), a novel technique to decrease noise by decreasing payload size. Adopt Low Rank Adaptation (LoRA) and Noise Contrastive Estimation (NCE) to reduce the memory demands of large models on compute-constrained devices. Then they utilize search to find a mask to ensure K-anonymity in this model. Outperforms masking based on named entities and matching with tabular data. This approach achieves high levels of privacy with low levels of redaction. Use RoBERTa-base and PMLM for document encoder & RoBERTa-base and TAPAS for table encoder
4. Unsupervised Deidentification of text - The approach learns to reidentify from text using a prior masking model.
5. Differential Privacy Meets Neural Network Pruning
6. Differentially Private Bias-Term only Fine-tuning of Foundation Models
7. Also idea of split learning (EXACT: Extensive Attack for Split Learning)
8. `https://github.com/lxuechen/private-transformers.`
9. `https://github.com/huseyinatahaninan/Differentially-Private-Fine-tuning-of-Language-Models`

A.7 Investigating LMs as Privatising Tools

NHS/Healthcare Scenario where this is Relevant For deidentifying health text for model training, identifying samples which would be memorized during model training and adjusting these or for deidentifying clinical model outputs.

BackgroundLarge Language Models can be used as a tool to assess and or remove private information content from text sequences. A novel approach learns to reidentify text from a prior masking model, then a search algorithm is used to find a mask which ensures K-anonymity of the example [88]. This method outperforms masking based on named entities and matching with tabular data. This approach achieves high levels of privacy with low levels of redaction. They use RoBERTa-base and PMLM for document encoder & RoBERTa-base and TAPAS for table encoder. In another study [89], a context-aware sensitive detection LM is trained, whose goal is to recognize sequences containing sensitive tokens. The detection LM is then applied to a training corpus and DP-SGD is only applied to sequences detected as sensitive. However, this method relies on the idea that secrets are in a standard format and can be easily identified.

Resources

1. Unsupervised Deidentification of Text
2. Context-Aware Differential Privacy for Language Modeling
3. A COMPARATIVE EVALUATION OF TRANSFORMER MODELS FOR DE-IDENTIFICATION OF CLINICAL TEXT DATA
4. `https://github.com/jxmorris12/unsupervised-text-deidentification`

A.8 Investigating Privacy of Reasoning in LMs

NHS/Healthcare Scenario where this is Relevant LM can reason over a medical/health problem in multiple steps whilst self-evaluating e.g. for patient triage.

Background LLMs have been shown to be able to perform a wider range of tasks, including using mathematical, commonsense and knowledge-based reasoning. The simple autoregressive (left to right) text generation mechanism underlies this progress but does not perform well when the mapping of input to output is non-trivial. Chain of thought (CoT) [90] prompting instead introduces a chain of thoughts to bridge the input and output where the step is a coherent language sequence serving as a meaningful intermediate step toward problem-solving. This can be considered as a maths problem where each intermediate step is an equation towards the final answer (or final output).

Tree of thoughts (ToT) [91] prompting builds upon CoT, actively maintaining a tree of thoughts, where each thought is a coherent language sequence that serves as an intermediate step toward problem-solving. LMs can perform intermediate deliberate decision-making by considering multiple reasoning paths and self-evaluating the choices to decide the next action. Search algorithms, such as breadth-first search (BFS) or depth-first search (DFS), are used to evaluate diverse thoughts, which allow systematic exploration of the tree of thoughts with lookahead and backtracking.

Causal reasoning can also be incorporated into LLMs [92] for example, methods to collect knowledge to generate causal graphs or identify background causal context from natural language. See existing causal methods are promising tools for LLMs to formalize, validate, and communicate their reasoning, especially in high-stakes scenarios.

Resources

1. Tree of Thoughts: Deliberate Problem Solving with Large Language Models
2. Towards Reasoning in Large Language Models: A Survey
3. Causal reasoning in LLMs
4. Alleviating Privacy Attacks via Causal Learning - not a LLM paper but shows models learnt using causal structure generalize better to unseen data, especially on data from different distributions than the train distribution establish a theoretical link between causality and privacy: compared to associational models, causal models provide stronger differential privacy guarantees and are more robust to membership inference attacks. Experiments on simulated Bayesian networks and the colored-MNIST dataset show that associational models exhibit up to 80% attack accuracy under different test distributions and sample sizes. In contrast, causal models exhibit attack accuracy close to a random guess.
5. Method enables models to decompose multi-step problems into intermediate steps.
6. Also Language Models (Mostly) Know What They Know -> get models to propose answers and then evaluate the probability that these are correct
7. Can large language models reason about medical questions?
8. `https://vlievin.github.io/medical-reasoning/`
9. `https://github.com/vlievin/medical-reasoning`

A.9 Investigating Privacy in Retrieval Augmented LMs

NHS/Healthcare Scenario where this is Relevant Practical example [93] - authors inject knowledge during training and into prompts. Something to note - knowledge graphs are often not the source of private data in medicine (often open ontologies etc. e.g. UMLS).

Background LLMs are black-box models that often falling short of capturing and accessing factual knowledge. In contrast, Knowledge Graphs (KGs), are structured knowledge models that explicitly store rich factual knowledge. KGs can enhance LLMs by providing external knowledge for inference and interoperability, termed KG-enhanced LLMs. KGs can not only be incorporated into the pre-training and inference stages of LLMs to provide external knowledge but can also be used for analyzing LLMs and providing interpretability.

Research on KG-enhanced LLMs includes using KGs: (1) during the pre-training stage and improve the knowledge expression of LLMs, 2) during the inference stage of LLMs, which enables LLMs to access the latest knowledge without retraining, 3) to understand the knowledge learned by LLMs and interpret the reasoning process of LLMs.

It has been demonstrated that a LMs ability to answer a fact-based question relates to how many documents associated with that question were seen during pre-training [75]. The authors demonstrate larger models are better at learning long-tail knowledge. Still, they estimate that today's models must be scaled by many orders of magnitude to reach competitive QA performance on questions with little support in the pre-training data. They show that retrieval augmentation can reduce the dependence on relevant document count, presenting a promising approach for capturing the long tail.

Privacy in KG-enhanced LMs has been investigated [76]. The authors trained GPT-2 a kNN-LM on Enron emails as the knowledge retrieval (key) model and used GPT-2-base as the query encoder. They find the KG-enhanced LM is more likely to leak private information vs the parametric model alone. They employ data sanitization and decoupling (query and key encoders) as mitigation to improve privacy preservation without sacrificing utility. They also consider training encoders on both public and private data to protect against untargeted attacks but find this does reduce utility by around 25%.

Resources

1. Unifying Large Language Models and Knowledge Graphs: A Roadmap
2. Privacy Implications of Retrieval-Based Language Models
3. Knowledge Enhanced PrompT (KEPT) framework