Did the Neurons Read your Book? Document-level Membership Inference for Large Language Models

Matthieu Meeus, *Imperial College London;* Shubham Jain, *Sense Street;*
Marek Rei and Yves-Alexandre de Montjoye, *Imperial College London*

Did the Neurons Read your Book?
Document-level Membership Inference for Large Language Models

Matthieu Meeus
Imperial College London

Shubham Jain
Sense Street

Marek Rei
Imperial College London

Yves-Alexandre de Montjoye
Imperial College London

Abstract

With large language models (LLMs) poised to become embedded in our daily lives, questions are starting to be raised about the data they learned from. These questions range from potential bias or misinformation LLMs could retain from their training data to questions of copyright and fair use of human-generated text. However, while these questions emerge, developers of the recent state-of-the-art LLMs become increasingly reluctant to disclose details on their training corpus. We here introduce the task of document-level membership inference for real-world LLMs, i.e. inferring whether the LLM has seen a given document during training or not. First, we propose a procedure for the development and evaluation of document-level membership inference for LLMs by leveraging commonly used data sources for training and the model release date. We then propose a practical, black-box method to predict document-level membership and instantiate it on OpenLLaMA-7B with both books and academic papers. We show our methodology to perform very well, reaching an AUC of 0.856 for books and 0.678 for papers (Fig. 1). We then show our approach to outperform the sentence-level membership inference attacks used in the privacy literature for the document-level membership task. We further evaluate whether smaller models might be less sensitive to document-level inference and show OpenLLaMA-3B to be approximately as sensitive as OpenLLaMA-7B to our approach. Finally, we consider two mitigation strategies and find the AUC to slowly decrease when only partial documents are considered but to remain fairly high when the model precision is reduced. Taken together, our results show that accurate document-level membership can be inferred for LLMs, increasing the transparency of technology poised to change our lives.

1   Introduction

Over the last year, Large Language Models (LLMs) have become ubiquitous. By understanding and producing coherent natural language, models such as GPT-2/3/4 [10, 47, 51],
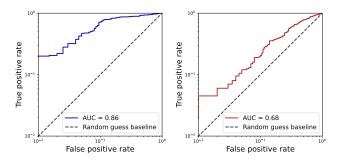
Figure 1: ROC curve for the best performing membership classifier (see Tables 1 and 3 for details). Results for books from Project Gutenberg (left) and ArXiv papers (right).

BLOOM [59] and LLaMA 1/2 [67, 68], promise to revolutionise society. ChatGPT, a fine-tuned version of GPT-3, was the fastest consumer-focused application in history to reach 100 million users [24]. Since this breakthrough, investment in Artificial Intelligence (AI) is estimated to reach $200 billion globally by 2025 [57].

While these models undoubtedly represent a major technical achievement, their capabilities stem from having been trained on enormous amounts of human-generated text. For instance, Meta's first generation model, LLaMA, had reportedly been trained on as many as 1.4 trillion tokens [67]. The capabilities of these models furthermore seem, at least at the moment, to keep improving with the size of the model (up to 100+ billion parameters) [7, 34]. This–in turn–fuels the need for increasingly more data, even leaving some wondering whether they might soon have consumed all the available data on the internet [71].

As LLMs become increasingly embedded into our daily life, simplifying tasks, generating value, but also making decisions, questions are being raised regarding the data from which their capabilities emerge. This includes questions about what they will learn, e.g. whether modern models will learn the societal biases present in the data or amplify them [1, 43, 62] or whether they might learn and propagate disinformation [5,

44, 66, 75]. Additionally, it also raises questions on ownership of the knowledge extracted by LLMs and whether copyright or *fair use* apply [58]. Indeed, content creators have raised concerns and filed lawsuits claiming copyright infringement by models trained on the books from the Books3 dataset, which would contain pirated copies of their content [25, 32], but also on songs [20] or news articles [45].

Unfortunately, as those questions are being asked, model developers are becoming increasingly reluctant to disclose and discuss their training data sources. One of the currently most widely used models, GPT-4, releases no information on its training dataset [47], while Meta initially released details on the training dataset of LLaMA [67], but now resists to do so in their release of LLaMA-2 [68].

Contribution. We here introduce the concept of document-level membership inference and propose what is -to the best of our knowledge- the first setup and methodology to infer whether an LLM has seen a document during training or not. Our method can help organizations, regulators, content creators and model users to evaluate whether a piece of content has been included in the training dataset of an LLM.

First, we formalize the task of document-level membership inference and propose a procedure and dataset for the development and the evaluation of document-level membership inference methodologies. The development of a meta-classifier, requires a dataset with *members*, documents that were likely seen by the model during training, and *non-members*, documents that are unlikely to have been seen by the model. Finding members is typically easy as most LLMs today have seen sources such as Common Crawl [16] or Project Gutenberg [26]. Given the large amount of data seen by LLMs today, finding *non-members* is more challenging. We thus propose to use documents similar to the one available in public sources but now made available after the LLM was released. We then construct a dataset to train and validate our inference methodology for books (Project Gutenberg [26]) and academic papers (ArXiv) and for OpenLLaMA [22], whose training dataset is known, RedPajama-Data [17].

Next, we introduce our methodology to infer, based on black-box access to the language model, whether a given document has been seen by a model or not. The methodology consist of the following steps: querying the model for token-level predictions, normalizing the predictions for how common the token is, aggregating the predictions to the document level, and lastly building a meta-classifier.

Our approach achieves an AUC of 0.856 and 0.678 for books and ArXiv papers respectively, on average. We also find that for this setup we retrieve a 64.4% and 19.5% true positive rate at a 10% false positive rate, for books and papers respectively, implying that our classifier can accurately identify members. Given the immense corpus used to train a state-of-the-art LLM such as OpenLLaMA [22], we find it remarkable that our auditor can distinguish the contribution of an individual document. Our results also show that normal-izing by both how common a token is and by the maximum probability returned by the model, is essential and that the signal for document-level membership lies in the detailed distribution of the token-level information within a document.

We then implement a sequence-level membership inference approach, such as previously considered in the privacy literature [14, 41, 73]. We show that state-of-the-art sequence-level baselines, scaled to document-level in three distinct ways, to perform poorly on our setup, reaching a maximum AUC of 0.56 and 0.57 for books and papers respectively.

Further, we investigate whether smaller models would impact the ability of an auditor to infer document membership. We find that even the smallest OpenLLaMA model, with 3 billion parameters, memorizes enough information about specific documents for the inference to be successful. In fact, our results show that the AUC remains at 0.86 and 0.68 for books and ArXiv papers respectively when switching from the 7B parameters to the 3B parameters models, indicating that even "smaller" models are at risk.

Finally, we consider potential membership inference mitigation strategies. We find that the AUC slowly decreases for partial documents, reaching an AUC of 0.62 for books when only 100 tokens are considered. For quantized versions of the language model, we still recover an AUC of 0.84 for books when the model is queried with *int4* precision.

2 Background

2.1 Language modeling

Recently, large language models (LLMs) have dominated the research in natural language processing [10, 51, 70]. Being trained in a self-supervised way on a large set of human generated text, LLMs are increasingly able to understand and generate natural language.

LLMs require a tokenizer T that discretizes a sequence of text to a sequence of n tokens $\{t_1, \ldots, t_n\}$. A token can be any sequence of characters that appear in language (e.g. a single word, part of a word or a combination of words). There is a finite number of tokens, and the collection of the tokens is referred to as the vocabulary $\mathcal{V}$ of size $V = |\mathcal{V}|$.

Given a sequence of n tokens, "foundation" language models are trained to optimize the probability for the model to generate a sequence of tokens $\{t_1, \ldots, t_n\}$, i.e. $p(t_1, \ldots, t_n)$. More specifically, auto-regressive language models like GPT-2 and GPT-3 [10, 51] are trained for "next-token prediction", i.e. to predict the likelihood of the next-token given the preceding tokens. These models have shown to improve understanding and particularly generation of natural language. They compute the probability of a sequence of tokens from a series of conditional probabilities as $p(t_1, \ldots, t_n) = \prod_{i=1}^{n} p(t_i | t_1, \ldots, t_{i-1})$.

The preceding sequence of tokens used to predict the probability of the next token is commonly referred to as the *context*, consisting of length $C = |\text{context}|$. Typically, language mod-

els have a maximum context length that can be taken into account, which we denote as C_{max}.

LLMs use neural networks to estimate this probability distribution, with a significant amount of parameters θ to be fit during training (up to 100+ billion). The predicted probability of LM with parameters θ for token t_i and context of length C can be formalized as $LM_\theta(t_i|t_{i-C},\ldots,t_{i-1})$.

The values for θ are optimized to maximize the likelihood of the sequence of tokens in a large dataset $\mathcal{D}_{\text{train}}$. $\mathcal{D}_{\text{train}}$ can for instance include a book or a Wikipedia page and in practice contains up to trillions of tokens. Here, we denote the total number of tokens in the training data as N_{train}. More formally, the model parameters θ are determined to minimize the following loss function: $\mathcal{L}(\theta) = -\log\prod_{i=1}^{N_{\text{train}}} LM_\theta(t_i|t_{i-C},\ldots,t_{i-1})$.

In the past, recurrent neural networks were the standard choice for the architecture of language models, but, over recent years, they have been replaced by attention-based models [70], in particular the transformer-based models, which now dominate the scene.

Since the release of ChatGPT, a fine-tuned version of GPT-3 optimized for chat-like behaviour, the development of ever better language models has further accelerated. Examples include Palm 1/2 [2, 15], BLOOM [59], LLaMA 1/2 [67, 68] and Mistral 7B [31]. At the same time, newly developed models are increasingly made publicly available on the open platform hosted by Hugging Face, which also deploys a LLM leaderboard to track and compare performances [1].

2.2 Datasets used for training

State-of-the-art LLMs consist of billions of parameters, which are trained on large-scale datasets [7, 34] containing trillions of tokens. Web-scraped data from the internet has long been the primary source for these large-scale datasets. Data retrieved from the frequently updated Common Crawl [16] has for instance been the majority of the training data for GPT-3 [10], LLaMA [67] and BLOOM [59].

Controlling the quality of the language included from these sources is however crucial. This led to the creation of curated datasets, such as WebText (which only contains text scraped from links sufficiently recognized on the social media platform Reddit, used for GPT-2) [51] and C4 [53] and the inclusion of more moderated content such as from Wikipedia.

Technology companies and researchers are in competition to develop ever better performing LLMs. The search for textual data, ideally of high quality, to be used to train the models is thus crucial. Exactly for this purpose, the training datasets have often been extended by other sources of high quality text, such as academic papers from ArXiv [37] or books from Project Gutenberg [26]. The latter contains thousands of English books in the public domain.

With the same objective of releasing high quality textual data, Goa et al [65] released the Pile, a 800GB dataset consisting of a diverse set of English text. This includes Books3, a dataset consisting of around 200,000 books obtained from pirate sources, most of which were published in the last 20 years and are thus not free of copyright [36, 55]. It is known that for instance BloombergGPT [72] and LLaMA [67] have been trained on Books3.

2.3 Copyright and generative AI

Content creators, such as authors or artists, have raised concerns about the inclusion of their work in the training data of generative AI, including LLMs but also multi-modal models such as DALLE-2 [54] and Stable Diffusion [56]. These concerns have led to multiple lawsuits against technology companies who have acknowledged the use of copyrighted material for training, without the consent of the creators.

For instance, Stability AI is currently defending against two lawsuits, one filed by Getty Images [23] and one filed as a class action lawsuit by multiple creators [3], both of which argue the use of copyright-protected content to train Stable Diffusion. Further, US comedian Sarah Silverman and other authors have filed lawsuits against Meta [32], claiming Meta has infringed their copyrights by training LLaMA on pirated copies of their content. The US Authors Guild also published an open letter, signed by more than 15,000 authors, calling upon AI industry leaders to protect their content [25]. Since then, other content creators have filed lawsuits against LLM developers claiming copyright infringement, including the New York Times [45] and Universal Music [20].

Notably, since these lawsuits and public concerns have emerged, the original data source for the Books3 dataset has been removed [2] and technology companies tend to not disclose details on the dataset used to train the latest language models any longer [31, 47, 68].

Recognizing these ongoing court cases, Samuelson [58] articulates the challenges between copyright laws and generative AI. In particular, they mention that the court will need to decide whether the inclusion of in-copyright works in the training data of AI models falls under *fair use*, in which case it would not be copyright-protected. While it is still uncertain how copyright applies to generative AI, it is clear that content creators are concerned [25] while the technology will continue to evolve rapidly without necessarily taking these concerns into account [8, 9].

2.4 Membership inference attacks

Homer et al. [28] introduced Membership Inference Attacks (MIAs) to study whether the presence of a certain target individual can be inferred from a genomic DNA mixture. They

---

[1] Hugging Face leaderboard

[2] https://huggingface.co/datasets/the_pile_books3

used statistical tests to distinguish between mixtures including the target and mixtures including individuals randomly drawn from a reference population. Since then, MIAs have been widely used to evaluate the privacy risk of aggregate data releases, such as location data [50] or survey data [6].

Since then, in the privacy literature, MIAs have also been developed against Machine Learning (ML) models [11, 19, 29, 64, 69]. Given a certain record D, a ML model trained on dataset $\mathcal{D}_{train}$, the attacker aims to infer whether the particular record was part of $\mathcal{D}_{train}$ or not. In many cases, the MIA setup makes additional assumptions on the access to the training data distribution and the ML model available to the attacker.

Shokri et al. [64] is seen as a foundational contribution in the field of MIAs against ML models. They assume the attacker to have black-box access to the model, i.e. they can query the model for predictions for a given input, and consider various assumptions on access to the training data. In their approach, the attacker uses the shadow modeling technique enabling them to learn a decision boundary for membership through a meta-classifier. Our proposed method also leverages a similar meta-classifier, inspired by the shadow modeling technique, while adapting it to our auditing setup.

MIAs against ML models have since been extensively studied in subsequent works and have become a common tool to assess what a ML model memorizes and whether its release poses any privacy risk [11, 18, 19, 29, 42, 65]. Importantly for this paper, MIAs have also been developed against (large) language models, which we discuss in the related work (Sec. 8).

3 Auditing setup

We consider an auditor $\mathcal{A}$ that has access to a document D, e.g. a book or a research article, and wants to check if D was used to train a language model LM. Thus, if we assume LM is trained using dataset $\mathcal{D}_{train}$, then the auditor wants to infer whether $D \in \mathcal{D}_{train}$ or not.

We define document D consisting of tokens t_i such that $i \in \{1,...,N\}$, or

$$D = \{SOS, t_1, .., t_i, .., t_N, EOS\} \tag{1}$$

where SOS is start-of-sequence token, and EOS is end-of-sequence token. Any such document D is highly likely to consist of significantly more tokens than the maximum context length of existing language models. Thus, we further assume that D consists of N tokens, and $N > C_{max}$.

We further assume that the auditor $\mathcal{A}$ has:

1. Query-only access to the language model LM and full access to its tokenizer T. This means that the auditor can query the model with a sequence of tokens S, and receive the probability output for all tokens v in vocabulary $\mathcal{V}$. This is a realistic assumption as trained models (along with their tokenizers) like LLaMA-2 [68] and Mistral

7B [31] are fully and freely released on platforms such as the one hosted by Hugging Face. We assume that the auditor is able to query LM an arbitrary number of times.

2. Access to two sets of documents, that stem from the same distribution as the document D e.g. if D is a book then D_M and D_{NM} also contain books. D_M and D_{NM} are defined as follows:

   (a) D_M: A subset of documents used in training of LM, also referred to as members, $\forall D \in D_M, D \in \mathcal{D}_{train}$.

   (b) D_{NM}: A subset of documents not used in training of LM, also referred to as non-members, i.e. $\forall D \in D_{NM}, D \notin \mathcal{D}_{train}$.

Note that access to D_M and D_{NM} is realistic in practice. First, most models use very similar datasets from sources such as Common Crawl [16] and Project Gutenberg [26], that are publicly available and easily accessible to include in D_M. Second, these sources are regularly updated and timestamped, which allows the auditor to construct D_{NM} by collecting the data that has been added after the training data has been collected (for which reasonable assumptions can be made as the model release is typically known). We leverage exactly this intuition to build the dataset for our experiments (Sec. 5.2). While not strictly required for the setup, we here consider a model for which each member document has been seen exactly once, and in its entirety, during training (Sec. 5.1).

Further, we assume that LM is a foundation model trained for next-token prediction on $\mathcal{D}_{train}$ which has not been subject to any measure that could shift the predicted probability distributions such as instruction tuning [74], reinforcement learning from human feedback [4] or watermarking [35].

Under these assumptions, the auditor aims to infer whether document D has been used to train the language model LM.

4 Methodology

The auditor $\mathcal{A}$ wants to build a meta-classifier $\mathcal{M}$ able to detect whether document D was used to train the language model LM. To achieve this, we extract document-level features for document D that could carry meaningful information on the membership of documents for the training of LM.

Intuition. From the privacy literature on MIAs against ML models [19, 64, 73], we learn that models tend to make more confident predictions on data samples that were seen during training than on data samples that were not. It is exactly this information that an auditor can leverage to infer membership. For LLMs, this confidence is reflected in the predicted probability for the true next token in the dataset given the preceding tokens. We use this intuition to construct the document-level features, which are then used as input for our meta-classifier $\mathcal{M}$. Specifically, our method considers the following four steps:

1. We first query *LM* to retrieve the predicted probability $LM(t_i|t_{i-C},\ldots,t_{i-1})$ of a token given the context of length C, and we do this for all tokens in document D, i.e. $\forall t_i \in D$. We try different values of C in our approach.

2. We then normalize the predicted probability of a token using a normalization algorithm NORMALIZE, for which we consider different options (Sec. 4.2).

3. We then consider multiple strategies FEATAGG to aggregate all the token level information for a document D to construct document level features.

4. Finally, the meta-classifier $\mathcal{M}$ takes as input the document-level features of D to make a binary prediction on membership of D.

We further formalize these four steps in the sections below.

4.1 Querying the model

The auditor queries *LM* to retrieve a value per token t_i in document D. As the maximum context length of the model C_{max} is likely to be smaller than the length of the document N, the auditor runs the model through the document with a certain fixed context length C and stride s.

The document is split in N_s sequences $S_j, j = 1,\ldots,N_s$ with $S_j = (t_{j,1},\ldots,t_{j,C})$ consisting of C consecutive tokens. The last sequence has C' tokens, which can be $\leq C$ depending on the total number of tokens N in D and stride s. We apply the same operations for this last sequence using C' instead of C.

Each sequence is passed to the model as input and results in a sequence of predicted probabilities for the corresponding true tokens:

$$LM_\theta(S_j) = \left\{ LM_\theta(t_{j,2}|t_{j,1}),\ldots LM_\theta(t_{j,C}|t_{j,1}\ldots t_{j,C-1}) \right\} \quad (2)$$

Note that $|LM_\theta(S_j)| = |S_j| - 1 = C - 1$, as the model does not return a prediction for the first token $t_{j,1}$ in the absence of a meaningful context to do so. Thus, when using context length C, we move through the document with stride $s = C - 1$. We then get sequences S_j for $j = 1\ldots N_s$ where $N_s = \lceil \frac{N}{C-1} \rceil$, so the resulting set of all $LM_\theta(S_j)$ contains predictions for all tokens t_i in D, except for the very first token t_1. We further refer to this probability of token t_i in document D for language model LM_θ simply as $LM(t_i)$.

The value for C is a fundamental hyperparameter of the setup and can be chosen to be any integer value smaller than C_{max}. Note that the predictions of a language model typically become more accurate for a longer context length C. However, as the goal is to retrieve information that should be meaningful for membership, it is not clear if this information is more likely to be at predictions with smaller or larger context.

4.2 Normalization (NORMALIZE)

In the section above, we query *LM* to extract $LM(t_i)$, i.e. the probability with which the *LM* would predict the next token

in question given a certain context. However, in natural language, some tokens are more rare than others. Intuitively, the probability with which a model predicts a certain token would depend on how rare such a token is. For instance, for a frequently occurring token such as *the* or *and* the model might predict a high probability, while for a more infrequent token the model naturally predicts a lower probability. We hypothesize that this occurs regardless of whether the model has seen the sequence of interest at training time or not.

Thus, to optimally retrieve the information meaningful for membership we *normalize* the model output with a value that takes into account the rarity of a token. Prior work on sequence-level membership inference has approached this in various ways. For instance, Carlini et al. [14] consider the zlib entropy of a sequence or use the predicted probability of a reference model. While zlib considers the entropy of a sequence and does not provide token-specific information, the use of a reference model requires using another model which also uses the exact same tokenizer T as *LM*, which is often not available. Instead, we propose different token-specific NORMALIZE approaches based on the data and the model the auditor $\mathcal{A}$ already has access to.

4.2.1 Computing a normalization dictionary

We consider the token $t_i = v$ in document D with v the token value part of the vocabulary $\mathcal{V}$. We then normalize the predicted probability of t_i, $LM(t_i)$, using reference value $R(v)$. $R(v)$ is calculated for each token v in vocabulary $\mathcal{V}$ to capture the rarity of v. We propose two ways of computing $R(v)$.

Token frequency (TF). Recall that the auditor is assumed to have access to both D_M and D_{NM}. With the concatenated dataset as reference $D_{ref} = D_M \cup D_{NM}$, the token frequency $R^{TF}(v)$ for token v in vocabulary $\mathcal{V}$ is then computed as follows:

$$R^{TF}(v) = \frac{\text{count}(v, D_{\text{ref}})}{\sum_{j=1}^{V} \text{count}(v_j, D_{\text{ref}})} \quad (3)$$

where $\text{count}(v, D_{\text{ref}})$ corresponds to the number of times token v appears in the set of documents D_{ref}. For tokens that do not appear even once in our dataset D_{ref} we use the smallest frequency in the normalization dictionary divided by 2 as a reference value.

General probability (GP). While the token frequency quantifies how rare a certain token is in the dataset, its computation does not take into account the specific context with which the token appears. Also, when a token v does not appear even once in D_{ref}, no valid value for the frequency $R^{TF}(v)$ can be computed.

In order to address both concerns, we propose to compute the *general probability* $R^{GP}(v)$ of the token v in vocabulary $\mathcal{V}$. Here, we run through D_{ref} as described in Sec. 4.1 and compute the average of all the predicted probabilities for v.

Note that we consider all predictions, also when the true token $t_k \in D_{\text{ref}}$ is not equal to v, as the model predicts the probability distribution over the entire vocabulary every time. $R^{GP}(v)$ is computed as follows:

$$R^{GP}(v) = \frac{1}{|D_{\text{ref}}|} \sum_{k=1}^{|D_{\text{ref}}|} LM_{\theta}(v \mid t_{k-C}, \ldots, t_{k-1}) \quad (4)$$

4.2.2 Normalization strategies

We here propose different normalization strategies NORMALIZE for $LM(t_i = v)$ using $R(v)$.

No normalization (*NoNorm*). In this case we use the predicted probabilities as they are and do not apply any normalization, or:

$$NoNorm(t_i = v) = LM(t_i) \quad (5)$$

Ratio normalization using the token frequency (*RatioNormTF*). We here compute the ratio of the predicted probability of the token of interest and the corresponding token frequency $R^{TF}(v)$ in D_{ref} as discussed in Sec. 4.2.1.

$$RatioNormTF(t_i = v) = \frac{LM(t_i)}{R^{TF}(v)} \quad (6)$$

Ratio normalization using the general probability (*RatioNormGP*). We here compute the ratio of the predicted probability of the token of interest and the corresponding general probability $R^{GP}(v)$ in D_{ref} as discussed in Sec. 4.2.1.

$$RatioNormGP(t_i = v) = \frac{LM(t_i)}{R^{GP}(v)} \quad (7)$$

Maximum probability normalization (*MaxNorm*). For a given token t_i, the model predicts the probability distribution over all tokens in the vocabulary $\mathcal{V}$. For now we have only considered the probability that the model returns for the true token t_i, regardless of how this probability compares to the predicted probability for the other token values with the same context. Intuitively, we could expect that the difference between the maximum predicted probability over all tokens $t \in \mathcal{V}$, with corresponding token $v_{\max}$, and the predicted probability for the true token t_i carries information about how likely the model is to predict the token of interest. Fig. 2 illustrates how the predicted probability for the true token and the maximum probability differ on a real example. We hypothesize that exactly the difference between both values could be meaningful to infer membership.

Formally, we denote the maximum probability being predicted for t_i for context C_i as

$$LM_{\max}(t_i|C_i) = \max_{t \in \mathcal{V}} LM(t|C_i) \quad (8)$$

Figure 2: Querying OpenLLaMA-7B on an example from the book *The Brothers Karamazov by Dostoyevsky* (member, Sec. 5.2).

regardless of whether the probability corresponds to the true token t_i or not. For rest of the paper, $LM_{\max}(t_i)$ implies $LM_{\max}(t_i|C_i)$. We then combine this with the ratio normalization strategies to get *MaxNormTF* and *MaxNormGP*.

$$MaxNormTF(t_i = v) = \frac{1 - (LM_{\max}(t_i) - LM(t_i))}{R^{TF}(v)} \quad (9)$$

$$MaxNormGP(t_i = v) = \frac{1 - (LM_{\max}(t_i) - LM(t_i))}{R^{GP}(v)} \quad (10)$$

Note that we ensure that the numerator never equals zero, even when the model predicts the highest probability for the true token t_i, or when $LM_{\max}(t_i) = LM(t_i)$.

4.3 Feature extraction (FEATAGG)

For all the tokens t_i in document D, we have so far normalized the predicted probabilities using normalization strategy NORMALIZE. We now use these normalized values to set $F(t_i)$ as the negative log of the output after normalizing token-level probabilities, or $F(t_i) = -\log(\text{NORMALIZE}(LM(t_i)))$.

Note that when we do not apply any normalization, or *NoNorm*, the value for $F(t_i)$ corresponds to the cross-entropy loss for the predicted probability, as used to optimize the model parameters during training.

While we now have computed a final value on the token level, our ultimate goal remains to predict binary membership for a document. Hence, from all token-level information $F(t_i), \forall t_i \in D$, we need to extract document-level features that could capture meaningful information for membership. We consider two feature extractors *AggFE* and *HistFE*.

Aggregate feature extractor (*AggFE*). For each document D, *AggFE* computes aggregate statistics from the token-level information. Specifically, it uses $F(t_i), \forall t_i \in D$ to compute the minimum, maximum, mean and standard deviation and the values at x-percentiles for $x \in X_{\text{perc}}$.

Histogram feature extractor (*HistFE*). *HistFE* uses N_b equal-sized histogram bins and for each document it computes the fraction of total values $F(t_i)$ in each bin. These N_b features are then used as document-level features. The N_b equal-sized bins are determined using all token-level values in the training dataset, across all documents.

4.4 Meta-classifier

As a final step, the document-level features extracted using FEATAGG are used as input to the meta-classifier $\mathcal{M}$, which returns a prediction for binary membership of input document D. As $\mathcal{M}$ we only consider a random forest classifier, allowing to fit non-linear dependencies across input features. We train $\mathcal{M}$ on a training subset of D_M and D_{NM} and evaluate its performance for binary classification on a disjoint subset.

5 Experimental setup

5.1 Model

As language model *LM* we use OpenLLaMA [22], a fully open-source reproduction of LLaMA [67], an auto-regressive LLM developed by Meta. While LLaMA is made publicly available for research and details on their training data have been provided [67], the exact training dataset has not been released. Instead, OpenLLaMA is trained on RedPajama-Data [17], which is a best-guess, open replication of the original dataset used to train LLaMA. We here opt for OpenL-LaMA in order to be in full control of the training dataset. The models are publicly available hosted by Hugging Face in three sizes: 3, 7 and 13 billion parameters. The tokenizer T has a vocabulary of size $V = 32,000$ and the maximum input length equals $C_{max} = 2048$.

5.2 Dataset for membership

We use two distinct types of documents, books and academic papers in LaTeX, to evaluate the effectiveness of our methods in inferring the document-level membership. For each type, we collect a fixed number of member (D_M) and non-member (D_{NM}) documents and train a separate meta-classifier.

As indicated in Sec. 3, we use realistic assumptions for the auditor to collect both D_M and D_{NM} for each document type. Specifically, for the members, we use data sources that are typically used for language modeling and are readily available. For non-members, we leverage the fact that these sources are regularly updated to retrieve the documents that have been added after the time the training data has been collected. While we believe these assumptions are reasonable in practice, we additionally ensure there is no overlap by using our exact knowledge of RedPajama-Data [17]. Below we describe in detail how we approach this for each document type.

5.2.1 Books (Project Gutenberg)

Project Gutenberg [26] is a volunteering project aiming to digitize and freely distribute books. The resulting online library contains thousands of books from a wide range of languages and publication dates. The vast majority of the releases are in the public domain in the United States, meaning that copyright laws do not apply.
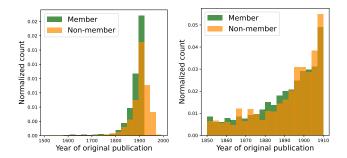
Figure 3: Density distribution of the original year of publication for books included in Project Gutenberg, for members and non-members. Both the raw distribution (left) and the filtered distribution for years 1850-1910 (right) are displayed.

Members D_M. We note that the RedPajama-Data includes both books from Project Gutenberg and the Books3 dataset made available by the Pile [21]. We exclusively consider the set of books PG-19 made available by Rae et al. [52]. This dataset contains 28,752 English books that were published before 1919 and made available on Project Gutenberg.

Non-members D_{DM}. Project Gutenberg is an ongoing project with new books added regularly. We create a comparable book dataset that has not been used for training by downloading books added to Project Gutenberg after the PG-19 dataset was created. Of all books included in PG-19, the latest release date on the Gutenberg project was February 10, 2019. We then use an open source library [49] to download all English books that were added to Project Gutenberg after this date. In our setup, this led to a total of 9,542 books that we could use as non-members. In line with how PG-19 has been constructed, we only consider the text between the explicit start and end of the uniformly formatted text files.

Two books published in different eras could be easily distinguished from the writing style. In that regard, Fig. 3 shows that there is a meaningful shift in year of original publication between books considered as member and non-members. Thus, there is a possibility that language written in books from for instance the 1700s can lead to data drift compared to books written in the 1900s. To ensure the meta-classifier $\mathcal{M}$ focuses on the memorization of the model, and does not focus on a potential drift in language, we only consider books with a year of original publication between 1850 and 1910. Fig. 3 shows that the distributions of members and non-members for these now filtered books are highly similar. This makes us confident that the model would focus on the membership rather than on the language drift.

5.2.2 Academic papers (ArXiv)

We use academic papers as posted on the online research-sharing platform ArXiv. In line with prior work training LLMs on academic papers [37], RedPajama-Data includes the raw

LaTeX text after removing the bibliography, the comments and everything before the occurrence of the first section.

Members D_M. As RedPajama-Data includes the preprocessed LaTeX text from ArXiv papers, we download documents that were part of the training dataset using their instructions [3]. This leads to approximately 1.5 million ArXiv papers to be considered as members of the training dataset.

Non-members D_{NM}. The Github repository of RedPajama-Data [17] also provides instructions on how to download academic papers from ArXiv directly. ArXiv maintainers host an Amazon S3 bucket containing all source LaTeX files, which gets updated on a monthly basis and can be retrieved at a moderate charge [4]. The data is grouped by the month on which the first version of the document has been added to ArXiv. From RedPajama-Data, we found that the last month that had been included in the training dataset was February 2023, which allows us to download all academic papers uploaded on ArXiv from March 2023 and use them as non-members. After applying the same preprocessing recipe as for the members, our non-member dataset of ArXiv paper eventually consists of around 155 thousand documents.

5.2.3 Training and evaluation setup

We use the documents considered as members and non-members to train and validate our method to predict membership of document D in the training data. To ensure our model focuses on documents of a reasonable length, and sufficiently longer than the maximum context length, C_{max}, we remove all documents, from both the books and academic papers, that have fewer than 5,000 tokens. We then randomly sample 1,000 member and 1,000 non-member documents. Next, we construct $k = 5$ different chunks of 400 documents, each chunk consisting of 200 member documents and 200 non-member documents. This allows us to train our set-up using $k - 1$ chunks, i.e. 1600 documents, and validate the performance on the held-out chunk, i.e. on 400 documents. By executing this k times, the classification performance is robustly estimated with a mean and standard deviation across k runs. For both document types, books and academic papers, we construct this setup to train and evaluate a distinct meta-classifier $\mathcal{M}$ k times.

5.2.4 Methodology parameters

We here describe the methodology parameters (Sec. 4.4) used in our experiments.

As context length C to query the model, we consider three different values $\{128, 1024, 2048\}$, the last of which is same as the maximum context length C_{max}. As D_{ref} to compute the normalization dictionaries in NORMALIZE, we use the same sets of documents used to train the meta-classifier. This means

that $k - 1 = 4$ chunks, with a total 800 members and 800 non-members are used, leading to $|D_{ref}| = 1,600$ for every run. For feature extractor *AggFE* we use a set of percentiles $X_{perc} = \{1, 5, 10, 25, 50, 75, 90, 95, 99\}$ and for *HistFE* number of bins $N_b = 1,000$. For the meta-classifier $\mathcal{M}$ we use a random forest classifier from scikit-learn [48] with 500 trees of a maximum depth of 5 and a minimum of samples per leaf of 3. We ran all language model queries on a set of A100 NVIDIA GPUs with a floating point precision of 16 bits.

6 Results

6.1 Evaluation across setups and datasets

Tables 1 and 3 summarize the performance of our binary membership classifier for books from Project Gutenberg and ArXiv papers respectively. We measure the performance of $\mathcal{M}$ using the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) and report the mean and standard deviation over $k = 5$ runs (Sec. 5.2.3).

The highest mean AUC achieved across setups for books and papers is 0.856 and 0.678 respectively, which is significantly higher than the random baseline of 0.5. This effectively shows that our model queries, normalization strategies and feature extraction enables the meta-classifier to learn the decision boundary for membership of a document D in the training data of an LLM. Around 250,000 books and 1.5 million ArXiv papers contribute to around 4.5% and 2.5% of the entire training dataset [67]. This makes the contribution of one document to the entire dataset negligible, and highlights the difficulty of membership inference. In this context, our meta-classifier's performance demonstrates the ability of our simple NORMALIZE and FEATAGG strategies to separate signal from the noise.

Further, the results allow us to compare the combinations of different setups, i.e. combination of context length C, normalization strategy NORMALIZE, and feature aggregation strategy FEATAGG as discussed in Sec. 4.4. First, we notice fairly little difference in performance across different values of context length C. This would imply that memorization is equally exposed for lower and larger values of context length. Second, we find that normalization is required to reach reasonable classification performances. Compared to the raw predicted probabilities *NoNorm*, all normalization strategies lead to an improved performance. Notably for *MaxNormTF*, we find that, for $C = 128$ and *HistFE*, the mean AUC increases by 0.29 and 0.10 for books and papers respectively when compared with *NoNorm* performance. Additionally, we find fairly little difference between normalizing with the *token frequency* $R^{TF}(v)$ or *general probability* $R^{GP}(v)$ across setups. This would imply that both normalization dictionaries provide similar information on the rarity of a token value. Third, we find that the histogram feature extraction *HistFE* is significantly more effective than the more simple aggregate statistics extraction

Table 1: Books from Project Gutenberg - mean and standard deviation AUC for binary membership across k folds.

FEATAGG	NORMALIZE	Context length C		
		128	1024	2048
AggFE	*NoNorm*	0.550 ± 0.011	0.551 ± 0.009	0.554 ± 0.012
	RatioNormTF	0.605 ± 0.025	0.552 ± 0.017	0.556 ± 0.012
	RatioNormGP	0.621 ± 0.031	0.539 ± 0.022	0.553 ± 0.021
	MaxNormTF	0.620 ± 0.029	0.546 ± 0.021	0.543 ± 0.016
	MaxNormGP	0.626 ± 0.025	0.542 ± 0.025	0.553 ± 0.021
HistFE	*NoNorm*	0.566 ± 0.022	0.528 ± 0.024	0.541 ± 0.017
	RatioNormTF	0.766 ± 0.030	0.786 ± 0.028	0.799 ± 0.020
	RatioNormGP	0.779 ± 0.010	0.790 ± 0.013	0.804 ± 0.011
	MaxNormTF	$\mathbf{0.856 \pm 0.021}$	0.854 ± 0.026	0.855 ± 0.027
	MaxNormGP	0.849 ± 0.019	0.853 ± 0.0218	0.852 ± 0.026

AggFE. For *MaxNormTF* and $C = 128$, *HistFE* leads to an increase in AUC of 0.24 and 0.05 for books and papers respectively compared with *AggFE* in an equivalent setup. This makes us confident that the information for membership truly lies in the detailed distribution of token-level values per document, rather than in high-level aggregates. Overall, $C = 128$, normalization using *MaxNormTF*, and feature aggregation using *HistFE* results in the best performing meta-classifier $\mathcal{M}$ for both books and academic papers.

	TPR@FPR	
Dataset	10%	1%
Project Gutenberg	$64.44 \pm 9.32\%$	$18.75 \pm 3.82\%$
ArXiv	$19.50 \pm 0.94\%$	$5.92 \pm 2.13\%$

Table 2: True positive rates at low false positive rates for the best setup in Tables 1 and 3.

Fig. 1 shows the ROC curve for one trained meta-classifier $\mathcal{M}$, randomly selected out of the k, for the best performing setup. In line with Carlini et al. [11], we also provide the true positive rate at low false positive rates for this setup in Table 2. Especially for the books, a true positive rate of 18.75% at a false positive rate of 1% implies a meta-classifier $\mathcal{M}$ that can confidently identify members in a given set of documents.

6.2 Comparison to sequence-level baseline

In the privacy literature, prior work on Membership Inference Attacks (MIAs) against language models has exclusively focused on inferring membership at the sequence-level [11,14,41,73]. In contrast, our setup concerns an auditor who aims to infer document-level membership.

We here compute how state-of-the-art sequence-level MIAs perform in our setup when scaled to the document-level. We consider as MIA methodologies to compute a sequence-level membership score α:

1. The *Loss* attack [73], which uses the language model cross-entropy loss computed for the given sequence.

2. Both the *Zlib* attack and *Lower* attack [14], which divides the target language model loss by the sentence zlib compression entropy and the target language model loss computed with all lower-case characters.

3. The *Ratio* attack [14], which divides the target language model loss for a given sample by the loss computed using a reference model. As a reference model, we here use both GPT-2 [51] (*Ratio-GPT-2*) and the OpenLLaMA-3B [22] (*Ratio-3B*).

4. The *Neighborhood* attack [41]. For each sample, we use a RoBERTa-based masked language model [39] to sample 50 neighboring samples with 1 token replacement. We use the difference between the target language model loss computed on the sample and the mean loss of the neighboring samples.

We now consider whether the sequence-level MIA can be scaled to the document-level. First, we split each document D in sequences $S_j = (t_{j,1}, \ldots, t_{j,C})$ of the same length $C = 128$. We found 1000 tokens to approximately map to 750 words, and hence using $C = 128$ tokens is a good proxy for selecting a sentence. For each document D, we randomly sample 40 sequences S_j, ensuring each document has the same number of sequences and then use the same membership label for sequences as for the corresponding document. We consider three ways of scaling the sequence-level membership inference to the level of the document:

1. *Sequence*. We compute the AUC on the sequence-level membership scores directly.

2. *Document (average)*. We use the average sequence-level

Table 3: ArXiv papers - mean and standard deviation AUC for binary membership across k folds.

FEATAGG	NORMALIZE	Context length C		
		128	1024	2048
AggFE	*NoNorm*	0.617 ± 0.015	0.605 ± 0.031	0.602 ± 0.022
	RatioNormTF	0.609 ± 0.027	0.614 ± 0.027	0.613 ± 0.017
	RatioNormGP	0.605 ± 0.028	0.609 ± 0.019	0.616 ± 0.014
	MaxNormTF	0.630 ± 0.021	0.622 ± 0.026	0.623 ± 0.026
	MaxNormGP	0.635 ± 0.016	0.632 ± 0.022	0.626 ± 0.023
HistFE	*NoNorm*	0.579 ± 0.029	0.580 ± 0.026	0.580 ± 0.028
	RatioNormTF	0.644 ± 0.028	0.647 ± 0.026	0.654 ± 0.033
	RatioNormGP	0.643 ± 0.031	0.638 ± 0.026	0.645 ± 0.030
	MaxNormTF	$\mathbf{0.678 \pm 0.024}$	0.668 ± 0.031	0.668 ± 0.031
	MaxNormGP	0.675 ± 0.019	0.665 ± 0.021	0.668 ± 0.029

membership score per document to compute the document-level AUC, in line with the group-level attack from [30].

3. *Document (threshold)*. We consider a threshold T for the membership score, below which a sequence is classified as member and non-member otherwise. We determine T so that a maximum classification accuracy is reached on the sequence-level on the training dataset. We then take the ratio of sequences within a document, i.e. $\forall (S_j) \in D$, that are predicted as member, to compute the document-level AUC.

Similarly to Sec. 6.1, we compute the AUC for membership on the sequence-level for k chunks. Table 4 summarizes the results, with the mean and standard deviation AUC for all MIA methodologies and strategies to scale to the document-level.

Overall, we find most baselines to barely perform better than a random guess, with the maximum AUC of 0.56 and 0.57 reached using *Ratio-3B* and *Lower* for books and papers respectively. This suggests that sequence-level MIAs from the privacy literature might not work out-of-the-box for LLMs trained on a significantly larger corpus than previously considered [11, 14, 41]. Further, this implies that using sequence-level information to predict document-level membership is possibly sub-optimal. Indeed, the signal for document membership might lie within the distribution of the token-level predictions rather than in the sequence-level predictions.

Comparison with concurrent work. Concurrently with this work, *Min-K% Prob* has been proposed as a new technique for sequence-level membership inference, which can also be scaled to detect document-level contributions to language model pretraining data [63]. Table 4 also summarizes the results for $K = 20$. For both datasets, we find *Min-K% Prob* to perform barely above a random guess baseline.

6.3 Evaluation across model sizes

Note that all results in the section above were achieved with the OpenLLaMA model with 7 billion parameters (7B) [22]. We here evaluate whether our membership inference methodology would work equally well for models with less parameters. We thus use $C = 2048$ and *HistFE* feature-aggregation strategy, against the model with 3B parameters. We consider the two best performing normalization strategies *MaxNormTF* and *MaxNormGP*, along with *NoNorm* for comparison.

Fig. 4 shows that the AUC for membership classification remains highly consistent for even the smallest model. This implies that the memorization, as measured by our meta-classifier, for a "smaller" model of 3 billion parameters remains fairly similar. While these findings contrast with prior work [12, 14], which show that memorization increases with model size, we note that our setup is significantly different. First, we only compare across larger models (3B+ parameters) and second, we measure memorization by evaluating document-level inference, which is very different than the extraction of specific sensitive information. Lastly, we also evaluate our setup using the model with 13 billion parameters and find that, again, the membership inference performance remains highly consistent.

6.4 Membership inference mitigations

Partial documents. All results above consider the documents in their entirety, leveraging all token-level probabilities. In some cases, the entire document might not be known to the auditor, or there could be limitations in how many times the LLM can be queried. Hence, we here evaluate how the membership inference performance changes when only partial documents are considered.

Table 4: Baselines - mean and standard deviation AUC for binary membership across *k* folds.

| Attack | Project Gutenberg | | | ArXiv | | |
	Sequence	Document (average)	Document (threshold)	Sequence	Document (average)	Document (threshold)
Loss	0.485 ± 0.012	0.478 ± 0.025	0.533 ± 0.016	0.530 ± 0.012	0.556 ± 0.027	0.563 ± 0.028
Zlib	0.453 ± 0.018	0.436 ± 0.025	0.541 ± 0.012	0.511 ± 0.007	0.522 ± 0.015	0.527 ± 0.015
Lower	0.505 ± 0.010	0.537 ± 0.024	0.520 ± 0.024	0.531 ± 0.009	$\mathbf{0.572 \pm 0.020}$	0.568 ± 0.022
Ratio-GPT-2	0.454 ± 0.011	0.407 ± 0.014	0.500 ± 0.009	0.505 ± 0.014	0.510 ± 0.025	0.530 ± 0.025
Ratio-3B	0.514 ± 0.006	$\mathbf{0.559 \pm 0.022}$	0.553 ± 0.024	0.485 ± 0.006	0.449 ± 0.022	0.489 ± 0.010
Neighborhood	0.508 ± 0.005	0.532 ± 0.025	0.530 ± 0.020	0.506 ± 0.010	0.513 ± 0.032	0.519 ± 0.025
*Min-K% Prob**	0.491 ± 0.008	0.486 ± 0.017	0.523 ± 0.020	0.531 ± 0.011	0.553 ± 0.024	0.557 ± 0.019

*Concurrent work.
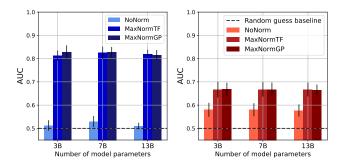
Figure 4: Mean AUC for $C = 2048$ across model sizes, for books from Project Gutenberg (left) and ArXiv papers (right).

Figure 5: Mean and standard deviation AUC for books from Project Gutenberg for an increasing number of tokens in a random excerpt sampled from the document.

Fig. 5 shows how the AUC varies for an increasing number of tokens taken into account for books from Project Gutenberg using the best setup from Table 1. In order to cleanly measure the impact of the number of tokens, we take a maximum number of tokens of $25,000$ and only consider books that have at least this number of tokens, eliminating 13.2% of the books. For a given number of tokens, a random excerpt of this length is sampled from the book and then considered as the full document to run the membership inference.

We find that even for 100 tokens a mean AUC of 0.62 is achieved, showing that even for smaller paragraphs sampled from documents our proposed method performs better than a random guess baseline. The AUC however steadily increases for an increasing number of tokens up until 0.83 for $25,000$ tokens, effectively showcasing how considering a larger part of the document can significantly increase the membership inference performance.

Quantized models. All results above consider the OpenLLaMA-7B in floating point precision of 16 bits. We here consider querying the language model with varying precision and evaluate the membership inference performance for books using the best setup from Table 1. From the results

in Table 5, we observe similarly good membership inference performance when querying the language model with lower precision. We hypothesize that the signal for membership picked up by our method likely does not change drastically with model precision when aggregating predicted probabilities over often more than $100,000$ tokens.

Precision	AUC
int4	0.841 ± 0.025
int8	0.835 ± 0.023
float16	0.856 ± 0.021
float32	0.843 ± 0.028

Table 5: Mean and standard deviation AUC for the best setup (Tab 1), querying OpenLLaMA-7B with varying precision.

6.5 Performance difference between datasets

The results in Tables 1 and 3 show a significant difference between the document-level membership inference performance on the books versus papers, with the highest mean AUC across setups of 0.856 and 0.678, respectively. We here discuss multiple hypotheses for this difference.

First, the data itself is inherently different. The papers from ArXiv consist of data from raw LaTeX files. This contains a highly specific set of characters (e.g. LaTeX formatting, table content) while the natural language included in books from the literature is expected to contain a wider diversity of tokens. Fig. 6 (a) compares how the token frequency R^{TF} is distributed across the top 20,000 tokens between papers and books. This confirms our hypothesis that LaTeX papers contain more of a limited set of tokens, while the tokens used in books are more widely spread over the entire vocabulary $\mathcal{V}$, possibly affecting the distribution of predicted probabilities that our method requires as input.

Second, academic papers and books have a different length, i.e. total number of tokens per document. Fig. 6 (b) confirms that the books considered in our experiments consist of significantly more tokens than the ArXiv papers, with an average of approximately 112,000 and 19,000 tokens respectively. This means that the language model has seen more tokens from member-books than from member-papers. Moreover, books typically contain repeated occurrences of characters, e.g. *Harry Potter*, across the document. These characters would likely be rare tokens, occurring more often for the longer books than for papers, likely impacting the model memorization more for books than for papers and thus also the classification performance. Further, the document length could also play a role at inference time of the meta-classifier. Indeed, during feature extraction, more rare tokens contribute to our final feature set in the case of books than for papers, which again would impact the classification performance.
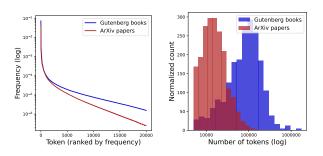
Figure 6: (a) Distribution of token frequency R^{TF} for the top 20,000 tokens. (b) Density distribution of number of tokens per document. Both contain results for 2,000 documents.

Third, we anticipate that books in Project Gutenberg often represent the pieces of literature that are wide-spread across the internet. This means that there might exist an overlap between the books as included in the PG-19 dataset for training and the data scraped from the internet such as Common Crawl [16] and C4 [53] which are also included in the training of OpenLLaMA [22]. This leads to a potential level of duplication, which is reasonably more likely for books that have been added to Project Gutenberg earlier than later (i.e. members). This will likely impact the memorization of these books in *LM* while, in contrast, academic papers, especially in LaTeX, are rarely distributed widely.

7 Discussion

We here construct a novel setup for document-level membership inference for real-world LLMs.

First, we introduce a general paradigm to construct a labeled dataset for document-level membership that we believe to be realistic in practice. Indeed, LLMs typically use similar data sources to construct a training dataset, such as Common Crawl [16], C4 [53] and Project Gutenberg [26]. This makes the identification of member documents fairly feasible in practice. Moreover, these sources are typically continuously updated and timestamped, which enables an auditor to retrieve similar documents that were added after the LLM training dataset was created, i.e. a comparable set of non-members.

This, however, depends on how recently a model was released and how frequently the data source of interest is updated. For ArXiv, more than 10,000 new papers are added monthly [5]. Hence, the retrieval of a sufficient amount of non-members (we here consider 1,000) should be feasible shortly after the model release, especially as training data is typically collected some time prior to the model release. For Project Gutenberg, we find that a handful of new books are added daily [6]. While less frequent, a reasonable amount of non-members could still be collected not too long after the model release.

Using this paradigm, we created a dataset of members and non-members for OpenLLaMA [22] that we could verify using the publicly available RedPajama-Data [17]. We leave for future work how the meta-classifier performance changes when the full knowledge of training data is not available. In particular, our setup could be applied to the original LLaMA [67] model and potentially even on LLaMA-2 [67].

In Sec. 6.3 we demonstrate the effectiveness of our method to language models of different model sizes (3B, 7B and 13B parameters) separately. In practice, it is however common that the entire model 'family' is trained on exactly the same dataset. We hypothesize that the auditor could construct a meta-classifier that takes as input document-level features queried from models of different sizes to make a potentially more accurate prediction for membership in practice.

In Sec. 6.4, we find that when the target language model is

---

queried with less precision, the membership inference remains accurate. We note, however, that we here exclusively consider pre-trained models, and leave for future work how alignment methods [4, 74] or watermarking [35], when applied to the language model, could alter these results.

Further, we constrained ourselves to use a dataset of members D_M and non-members D_{NM} of size 1000 each. With our computational resources, it takes us approximately a day to generate the predicted probabilities for each token in the books dataset by querying language model LM. We expect that researchers with a large resource pool could explore the impact of having larger datasets of members and non-members on the performance of the meta-classifier $\mathcal{M}$.

Lastly, with this proof-of-concept, we hope to provide a way to retrieve a reasonable estimate of whether a document has been included in the training dataset of an LLM. Not only does this improve our understanding of memorization in ever larger models in practice, it also encourages increased transparency in what LLMs are trained on.

8 Related work

There exists a significant literature focusing on privacy attacks against (large) language models, which are relevant to our setup for document-level membership inference.

Carlini et al. [13] proposes a framework to quantitatively asses unintended memorization in generative sequence models, such as language models trained for next token prediction. They effectively showcase how log-perplexity from a LSTM recurrent neural network trained for next token prediction can be used to extract potentially sensitive sequences from the training data (such as emails or credit card information). Further, Song et al. [65] proposes a method to audit text generation models. Specifically, they use the shadow modeling technique to train a binary classifier for membership of the training data. For text generation, they implement this on the user level, aggregating predictions from individual pieces of text associated with the same user. While conceptually similar to the document-level membership inference in this paper, Song et al. apply their approach on recurrent neural networks trained on small datasets. Due to the computational cost to train a state-of-the-art, real-world large language model [10,31,51,67,68], the use of the shadow modeling technique is no longer feasible. Further, the size of the dataset used to train these models [7] has drastically increased, making the contribution of an individual document distinctly harder to distinguish. For both reasons, new methods such as we here propose, are required to infer document-level membership against the most recent models.

Subsequently to their prior work, Carlini et al. [14] introduces a novel method for training data extraction from the transformer-based GPT-2, with up to 1.5 billion parameters trained on a vast training dataset of 40GB of text [51]. Importantly for this paper, they show that a combination of per-plexity queried from the model and zlib entropy (which gives a certain reference notion of surprise for a certain sequence of characters) allows for effective extraction of training data. Additionally, they show that larger models, i.e. neural networks with more parameters, tend to memorize more from their training data. Mattern et al. [41] proposes a neighborhood attack for MIAs against language models. They also attack GPT-2, although now fine-tuned on specific data for which membership is then inferred. They use the difference between the target model loss computed on a given sequence and the loss computed on neighboring samples, obtained by replacing tokens using masked language models.

In contrast with the prior contributions mentioned above, this paper focuses specifically on document-level membership inference and applies it to real-world, large language models (7B+ parameters). This is, to the best of our knowledge, the first work of its kind.

Further, researchers have proposed defenses against these privacy attacks, such as de-duplication of the training data [33] or differentially private training [38]. However, Lukas et al. [40] challenges this, stating that these defenses do not reduce the risk posed to personal identifiable information. Lastly, various other works have developed membership inference attacks on more domain as well as task-specific language models, such as clinical language models [30] or language models in non-English languages [46], or fine-tuned models for machine translation [27] or classification [60,61].

9 Conclusion

This paper proposes the first setup and methodology for document-level membership inference for real-world LLMs.

First, we formalize the task and introduce a procedure to realistically construct a labeled training dataset. For non-members, we rely on original documents that are similar to documents seen by the model but made available after the release date of the model. We then construct a dataset for the OpenLLaMA [22] model for both books from Project Gutenberg [26] and academic papers from ArXiv.

We then propose a methodology to infer document-level membership. We find that the distribution of predicted probabilities for all the tokens within a document, normalized by a value reflecting the rarity of the token, contains meaningful information for membership. Indeed, in the best performing setup, the meta-classifier infers binary membership with an AUC of 0.86 and 0.68 for books and papers respectively. This suggests our classifier's ability to accurately infer whether a document has been seen by the LLM during training.

Availability

The code used to generate the results in this paper has been made publicly available on Github[7].

References

[1] Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 298–306, 2021.

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] Artists v stability ai. bloomberg-law-stability, 2023.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[5] Francois Barnard, Marlize Van Sittert, and Sirisha Rambhatla. Self-diagnosis and large language models: A new front for medical misinformation. *arXiv preprint arXiv:2307.04910*, 2023.

[6] Luke A Bauer and Vincent Bindschaedler. Towards realistic membership inferences: The case of survey data. In *Annual Computer Security Applications Conference*, pages 116–128, 2020.

[7] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[8] Blake Brittain. Meta tells court ai software does not violate author copyrights. reuters-meta-tells-court-ai-software-does-not-violate-author-copyrights, 2023.

[9] Blake Brittain. Openai asks court to trim authors' copyright lawsuits. openai-asks-court-trim-authors-copyright-lawsuits, 2023.

[10] 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.

[11] 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.

[12] 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.

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

[14] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[15] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[16] Common crawl. https://commoncrawl.org.

[17] Together Computer. Redpajama-data: An open source recipe to reproduce llama training dataset, April 2023.

[18] Ana-Maria Cretu, Daniel Jones, Yves-Alexandre de Montjoye, and Shruti Tople. Re-aligning shadow models can improve white-box membership inference attacks. *arXiv preprint arXiv:2306.05093*, 2023.

[19] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

[20] FinancialTimes. https://www.ft.com/content/0965d962-5c54-4fdc-aef8-18e4ef3b9df5, Oct 2023.

[21] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

---

[7] https://github.com/computationalprivacy/document-level-membership-inference

[22] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023.

[23] Getty images v stability ai. `https://aboutblaw.com/6DW`, 2023.

[24] Cindy Gordon. Chatgpt is the fastest growing app in the history of web applications. forbes-chatgpt-is-the-fastest-growing-app, 2023.

[25] Authors Guild. More than 15,000 authors sign authors guild letter calling on ai industry leaders to protect writers. authors-guild-open-letter, 2023.

[26] Michael Hart. Project gutenberg.

[27] 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.

[28] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[29] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

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

[31] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[32] LLP JOSEPH SAVERI LAW FIRM. Kadrey, silverman, golden v meta platforms, inc. `https://llmlitigation.com/pdf/03417/kadrey-meta-complaint.pdf`, 2023.

[33] 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.

[34] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[35] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

[36] Kate Knibbs. The battle over books3 could change ai forever. wired-battle-over-books3, 2023.

[37] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

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

[39] 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.

[40] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE Computer Society, 2023.

[41] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

[42] Matthieu Meeus, Florent Guepin, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Achilles' heels: Vulnerable record identification in synthetic data publishing. *arXiv preprint arXiv:2306.10308*, 2023.

[43] Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*, 2020.

[44] Bloomberg News. openai-chatbot-so-good-it-can-fool-humans-even-when-it-s-wrong. openai-chatbot-so-good-it-can-fool-humans-even-when-it-s-wrong?, 2022.

[45] NewYorkTimes. The times sues openai and microsoft over a.i. use of copyrighted work. `https://`

www.nytimes.com/2023/12/27/business/media/
new-york-times-open-ai-microsoft-lawsuit.
html, Dec 2023.

[46] Myung Gyo Oh, Leo Hyun Park, Jaeuk Kim, Jaewoo Park, and Taekyoung Kwon. Membership inference attacks with token-level deduplication on korean language models. *IEEE Access*, 11:10207–10217, 2023.

[47] OpenAI2023. Gpt-4 technical report. https://cdn.openai.com/papers/gpt-4.pdf, 2023.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[49] Katherine Pully. Gutenberg scraper. https://github.com/kpully/gutenberg_scraper, 2020.

[50] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*, 2017.

[51] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[52] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

[53] 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.

[54] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[55] Alex Reisner. These 183,000 books are fueling the biggest fight in publishing and tech. the-atlantic-books3-copyright, 2023.

[56] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[57] Goldman Sachs. Ai investment forecast to approach 200 billion globally by 2025. goldman-sachs-ai-investment, 2023.

[58] Pamela Samuelson. Generative ai meets copyright. *Science*, 381(6654):158–161, 2023.

[59] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[60] Shlomit Shachor, Natalia Razinkov, and Abigail Goldsteen. Improved membership inference attacks against language classification models. *arXiv preprint arXiv:2310.07219*, 2023.

[61] 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.

[62] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*, 2019.

[63] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[64] 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.

[65] 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.

[66] New York Times. Disinformation researchers raise alarms about a.i. chatbots. nytimes-ai-chatbots-disinformation, 2023.

[67] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[68] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov,

Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[69] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019.

[70] 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.

[71] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 2022.

[72] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[73] 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.

[74] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

[75] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.