Master in Intelligent Interactive Systems
Universitat Pompeu Fabra

The Impact of Training Order on Memorization and Forgetting in Large Language Models

Danny D. Leybzon

Supervisor: Corentin Kervadec

July 2024

Universitat
Pompeu Fabra
Barcelona

Contents

Abstract

This paper investigates the phenomenon of memorization in large language models (LLMs), focusing on its dynamics throughout the training process and its implications for data privacy and copyright compliance. While LLMs have demonstrated impressive performance across various natural language processing tasks, their tendency to reproduce portions of their training data verbatim raises concerns about the potential leakage of sensitive information and copyright infringement.

Our research reveals several key findings about the memorization process in LLMs. First, we observe that models tend to memorize a higher proportion of their training data during the early stages of training. This memorization rate exhibits logarithmic growth before stabilizing into a linear pattern. Notably, this logarithmic growth is attributed to an increase in the number of examples forgotten by the model at each step, while the number of newly memorized examples remains relatively constant.

We demonstrate that the dynamic nature of memorization results in few examples being retained throughout the entire training process. However, forgotten examples are often re-memorized during subsequent training steps. Importantly, examples memorized early in training have a higher likelihood of remaining memorized throughout the entire process.

Based on these findings, we tentatively recommend that model developers avoid including their most sensitive data at either the very beginning or end of the training process to mitigate potential risks associated with memorization.

Our study also reveals that different types of text are memorized at varying rates, although the overall memorization dynamics remain consistent across text categories. We find that contact information is disproportionately memorized, and many examples that persist in memory throughout training, especially those containing contact information, follow a "templated" structure.

These insights contribute to a deeper understanding of memorization in LLMs and provide a foundation for developing strategies to minimize the memorization of sensitive information and mitigate the risk of training data extraction attacks. Furthermore, our findings have implications for addressing copyright concerns in the development and deployment of large language models.

This research advances the field by offering a more nuanced view of memorization dynamics in LLMs and provides practical recommendations for model developers to enhance data privacy and copyright compliance in their training processes.

Keywords: Large Language Models; Memorization; Training

Chapter 1

Introduction

Large language models (LLMs) are capable of achieving state-of-the-art results on a variety of NLP tasks [1] but are not without their problems. One such problem is their propensity to output portions of their training data verbatim, a phenomenon referred to as "memorization" [2]. Although this memorization may have desirable effects (discussed in Section 2.1) , it also potentially causes two undesirable behaviors from LLMs: leaking sensitive data and violating copyright laws.

Memorization in LLMs is a potentially undesirable outcome because it can lead to the unintentional disclosure of private information such as personal data (including credit card or social security numbers), trade secrets, passwords, etc.[2] Training data extraction attacks seek to extract training examples from a model verbatim and memorization enables these types of attacks to succeed [3] [4]. By better understanding why memorization occurs, researchers will be able to minimize the memorization of sensitive information and mitigate the risk of training data extraction attacks[5].

Memorization is also potentially undesirable because it can lead language models to generate text that violates copyright law [6][7]. For instance, OpenAI has been sued by The New York Times for allegedly using New York Times articles in training their ChatGPT model, as evidenced by the model allegedly outputting articles nearly verbatim[8].

Previous work [9] (discussed in Section 2.2) has concluded that LLMs memorize a fixed proportion of their data at each step and, as a result, has avoided making recommendations about the order in which data is fed to the model throughout training. Our work shows that:

1. Models tend to memorize a higher proportion of their training data early on during training
2. That the memorization rate grows logarithmically until it flattens out and becomes linear
3. That this logarithmic growth is actually caused by a logarithmic growth in the number of examples forgotten by the model at each step, while the number of newly memorized examples stays nearly-constant
4. That this process of forgetting means that very few memorized examples are memorized throughout the entire training process
5. But that forgotten examples get re-memorized throughout training
6. That examples memorized early on in training are more likely to remain memorized throughout the entire training process

As a result, we tentatively recommend model developers to avoid putting their most sensitive data either in the beginning of the training process or at the very end.

In addition to these major conclusions, we also conclude:

1. That different types of text get memorized at different rates
2. But that the memorization dynamics throughout training are the same regardless of text category (according to the categorization schema we discuss in Section 3.5)
3. That contact information is disproportionately memorized
4. That many of the examples that remain memorized for most of training, especially the ones that contain contact information, follow a "templated" structure

Chapter 2

Background

2.1 Defining Memorization in Language Modeling

Early research on memorization in deep neural networks studied memorization through the lens of generalization, treating memorization as a case of overfitting which would result in poor generalization [10]. More recent work has shown that memorization does not result in diminished generalization [11].

The first research into memorization in LLMs specifically showed that LLMs are capable of acting as knowledge bases [12] and quantified the amount of knowledge that an LLM could retrieve from its training data [13]. While this information retrieval is a desirable quality of LLMs, other research explored *unintentional* memorization and studied memorization through the lens of information security [2]. Those authors describe "unintentional memorization" as the memorization of data that is irrelevant to the learning task. While this unintentional memorization may be undesirable when it happens on personal data, there is also theoretical evidence that it may be essential for LLM performance [14] and may be an important prerequisite for more robust learning in LLMs [15].

Further refining this definition, later authors [3] introduce a quantifiable definition of "k-eidetic memorization": A string s is k-eidetic memorized (for $k \geq 1$) by an LM f_θ if s is extractable from f_θ and s appears in at most k examples in the training data X:

$$|\{x \in X : s \subseteq x\}| \leq k.$$

Key to the definition of memorization is “extractability”, which refers [16] to the ability to prompt a model to generate a string given a text prompt of length k which precedes the target string in the training data. More concretely: A string s is *extractable with* k *tokens of context* from a model f if there exists a (length-k) string p, such that the concatenation $[p \parallel s]$ is contained in the training data for f, and f produces s when prompted with p using greedy decoding.

Note that the variable "k" is used differently in these two definitions. Note also that, although extractability is a necessary and sufficient condition for memorization, the definitions differ.

All strings that are extractable in such a way are counted as memorized. Indeed, extractability acts as a highly sensitive “canary in the coal mine” for other, more harmful forms of memorization, like the ones taken advantage of in training data extraction attacks [4]. If training data is extractable via prompting the model with extracts from its training data, it is possible that other attack vectors will also allow sensitive training data to be extracted.

2.2 Memorization Training Dynamics

Previous work on the training dynamics of memorization in language models has primarily been motivated by preventing memorization or getting early signals of it during training. Memorization rates have been found to scale with parameters such as model size [16][17][18], the frequency of appearance of the example in the dataset [16][19], the length of the context k used to prompt the model [16], and the learning rate [17]. Overall memorization rate (the percentage of training data that is memorized by the model) estimates vary widely based on the definition of memorization used.

Research into memorization training dynamics in vision models found that memorization occurred less frequently earlier in training and more frequently later [20]. On the other hand, research on memorization in language modeling has found that memorization is well-modeled by a Poisson distribution, indicating that memorization is approximately equally likely to happen at each step in the training process [9]. Further research found little correlation between the examples memorized throughout the training process, indicating that the model is forgetting many of the examples it had previously memorized and then re-learning them seemingly at random [18]. These findings are in contradiction to the phenomena that we observe in our analysis.

2.3 Forgetting in Language Modeling

Few studies have discussed "forgetting" in the context of LLM memorization research. Most memorization research we surveyed is not focused on the training dynamics of memorization and the ones focused on training dynamics [9][18] did not discuss forgetting. A notable exception is [17], where the authors find a logarithmic forgetting curve that ultimately comes to a stable "forgetting baseline", primarily dictated by model size.

There has been more prior work about forgetting in the context of LLM information retrieval. That work has focused on "catastrophic forgetting" (also referred to as "catastrophic interference"), which is where machine learning models forget previously learned information in the process of acquiring new information [21]. In that context, forgetting is an undesirable quality, and research has been performed through the framework of continual learning [22], the effects of fine-tuning [23], strategies to mitigate forgetting [24], and even forgetting dynamics during training [15]. Since those works focus on LLMs as knowledge bases, where "learning" is a desirable quality and therefore "forgetting" is undesirable, this view of forgetting is fundamentally different from forgetting in the context of memorization, where "memorization" is undesirable and "forgetting" is desirable.

2.4 OLMo

Our model of choice in this work is the 7 billion parameter Open Language Model (OLMo) [25] published by the Allen Institute for Artificial Intelligence. OLMo is a framework that consists of trained OLMo models, the pre-training dataset Dolma [26], and various other artifacts. The OLMo models are decoder-only LLMs that have been trained using similar practices to the currently available, state-of-the-art LLMs and are competitive with those LLMs in many of the OLMo authors' evaluations. This makes them an ideal proxy for evaluating memorization and forgetting in those state-of-the-art LLMs, which we can not evaluate directly because they do not follow the same open framework as OLMo.

We make use of the fact that OLMo models are published with not only final model weights but also their training dataset, training methodology, and intermediate checkpoint models. These intermediate checkpoints are evenly-spaced snapshots of the models' weights throughout the training process, which can be run as if they are fully trained models. These checkpoints allow us to evaluate memorization throughout the learning process, not just in the final model, which allows us to understand the dynamics of how examples get memorized and forgotten.

While other frameworks, such as the Pythia suite [9] and BLOOM [27], also release training data, methodology, and checkpoints, we choose to work with OLMo on this analysis because it is the most representative of state-of-the-art LLMs. It is trained on a much larger and more processed dataset, following current best practices, and performs comparably to similarly-sized models. We reproduce our OLMo analysis using a comparably-sized Pythia model in Section 7.2.

Chapter 3

Methodology

To study the impact of training order on memorization, we extracted and then deduplicated 64-token sequences from OLMo's training dataset. We then passed the first 32 tokens of these sequences to evenly-spaced checkpoints throughout OLMo's training process and had these checkpoints generate 32 more tokens. We compared these generated tokens with the "ground truth" (i.e. the last 32 tokens in the original extractions) to evaluate whether and to what extent the sequence had been memorized. Finally, we automatically labeled the examples that had been memorized by the final model as well as an equivalently sized sample of the whole dataset to understand qualitatively how the memorized examples differed from the non-memorized ones. See Figure 1.

3.1 Sequence Extraction

The version of OLMo used in this paper was trained on version `v1_5-sample` of the Dolma dataset [26]. This corpus is split into 2,418 files, each of which contains a list of documents sorted by their source. For each file, we extracted the first 500 documents that had a length greater than or equal to 64 tokens and extracted the first 64 tokens from each document. We chose to extract the first 64 to reproduce the work in [9], where the length 64 is chosen arbitrarily and the first tokens are extracted to minimize covariate effects. This resulted in a dataset of 1,208,000 sequences of

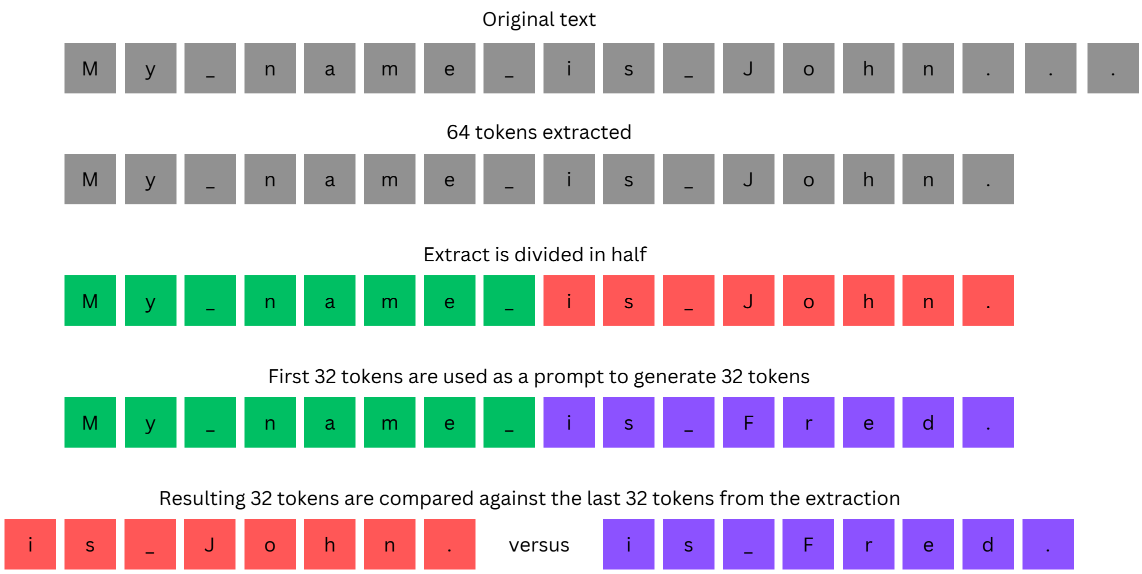

Figure 1: A simplified version of our methodology which uses characters instead of tokens.

length 64, where each sequence appeared at the beginning of a document in Dolma. Two files did not have any documents with lengths greater than 64, which explains the 1,000 sequence discrepancy between our final sequence count and the expected final sequence count of 1,209,000 ($2,418 * 500 = 1,209,000$).

3.2 Deduplication

Prior research has shown that repeated examples in the training data are more likely to be memorized [16][19][28][29]. Although the Dolma dataset that OLMo was trained on has been heavily deduplicated, some sequences repeat in various places in the training data. To minimize the impacts of often-repeated sequences on our analysis, we deduplicated our dataset before performing our analysis.

This resulted in a marginal decrease of 0.2%, implying that the duplication rate in the overall dataset is quite low. We further analyze duplication in Section 7.3.

3.3 Response Generation

After deduplicating our data, we split each sequence into two 32-token subsequences. We selected 112 checkpoints separated by 5,000 training steps each, starting from step0-tokens0B (which represents the randomly initialized model that has been exposed to no training data) to step555000-tokens2455B (which represents the fully-trained model that has been exposed to approximately 2,455,000,000 tokens). We passed the first 32-token subsequence prompts to the model and generated 32-token responses using greedy decoding, following the standard definition of extractability [16]. During generation, we used the default HuggingFace function parameters, except for using 16-bit quantized versions of the checkpoints and running the generations on our GPUs. We use prompt batches of size 32.

3.4 Memorization Evaluation

We evaluated whether a checkpoint had memorized a given sequence by directly comparing the 32-token sequence generated by the model against the original 32-token response we had extracted from the training dataset. If the generated sequence exactly matched the ground truth sequence, we counted that sequence as a "memorized" example for that checkpoint. We also evaluated other, novel definitions of memorization, as discussed in Section 7.1.

3.5 Classification

Previous work [3] manually categorized memorized examples into 16 categories:

1. US and international news
2. Log files and error reports
3. License, terms of use, copyright notices
4. Lists of named items (games, countries, etc.)
5. Forum or Wiki entry Valid URLs
6. Named individuals (non-news samples only)
7. Promotional content (products, subscriptions, etc.)

8. High entropy (UUIDs, base64 data)
9. Contact info (address, email, phone, twitter, etc.)
10. Code Configuration files
11. Religious texts
12. Pseudonyms
13. Donald Trump tweets and quotes
14. Web forms (menu items, instructions, etc.)
15. Tech news
16. Lists of numbers (dates, sequences, etc.)

We wanted to understand how the type of text might affect memorization and forgetting dynamics throughout the training process. We took the examples memorized by the final checkpoint and an equally-sized random sample of all sequences we evaluated, including both memorized and not memorized sequences. We used the same taxonomy (adding an "Other/Misc" category for examples that did not fall into any of these categories) and automatically classified these examples.

To automatically classify the memorized and sampled examples, we selected the top-performing open-source model on Chatbot Arena [30] that would fit on our GPUs: Meta-Llama-3-8B-Instruct [31]. We prompted it to automatically classify the examples, using the prompt specified in Subsection 7.4.1. We extracted these classifications, examples of which are available in Subsection 7.4.2. In a qualitative evaluation, we found these classifications to be imperfect, as discussed in Section 5.3.

Chapter 4

Results

4.1 Descriptive Statistics

Table 1: Descriptive statistics about our evaluation of memorization in OLMo.

Sequences evaluated	**Checkpoints used**	**Memorized by at least one checkpoint**	**Memorized by the final checkpoint**	**Memorized at every checkpoint**
1,205,572	112	44,559 (3.7%)	26,423 (2.19%)	1,127 (0.09%)

Of the 1,2080,000 sequences extracted from OLMo's training data in Sequence Extraction, 1,205,572 remained after deduplication. Of these, 44,559 were memorized by at least one of the 112 OLMo checkpoints we considered. This is equivalent to 3.7% of the sequences. The step0, randomized model had memorized sequences zero, while the final model had memorized 26,423 (2.19% of all sequences). There were 1,127 (0.09% of the total) sequences memorized at every checkpoint we evaluated, excluding the step0 checkpoint.

The memorization rate we observe in the final checkpoint (2.19%) is similar to the memorization rates reported in other works (e.g. over 1% for GPT-J [16]). The memorization rate is a function of many variables, including the length of the prompt used to extract a response [16], and thus we should not extrapolate raw memorizations rates of LLMs without specifying the corresponding prompt lengths. The facts that 59.3% of examples memorized by any checkpoint are memorized by the final checkpoint and that only 0.09% of examples are memorized by every checkpoint demonstrate an important insight in this work: LLMs memorize their training data but then forget parts of it throughout the training process. As further analysis will demonstrate (especially in Section 4.2 and Section 4.3) sometimes examples are memorized, forgotten, and then re-memorized again in subsequent checkpoints.

4.2 Final Checkpoint Analysis

Model developers and researchers may be particularly interested in understanding the examples that the final checkpoint (i.e. the model at the end of the training process) has memorized. This might be of particular interest because this checkpoint represents the model that will either be deployed directly to users or fine-tuned and then deployed. With that in mind, we start our analysis by looking at only examples memorized by the final checkpoint and seek to understand how and when they were memorized.

Of the 1,205,572 sequences we tested for the OLMo model, 44,559 were memorized by at least one checkpoint, but only 26,423 were memorized at the final checkpoint. These examples were not all memorized by the final checkpoint itself, but rather accumulated over the course of the training process. To understand this phenomenon, we start by plotting how many examples memorized at the final checkpoint were memorized at each prior checkpoint, as seen in Figure 2.

The plot contains three distinct sections, which is indicative of the memorization dynamics of the final model. The first 20% of the data display what appears to be logarithmic growth in the number of memorized examples at each consecutive checkpoint. For the last 80%, there appears to be fixed, linear growth in the number of memorized examples, with some noise. Finally, at the last checkpoint, there is a large spike in the number of memorized examples.

This provides early evidence for one of our conclusions: without futher information

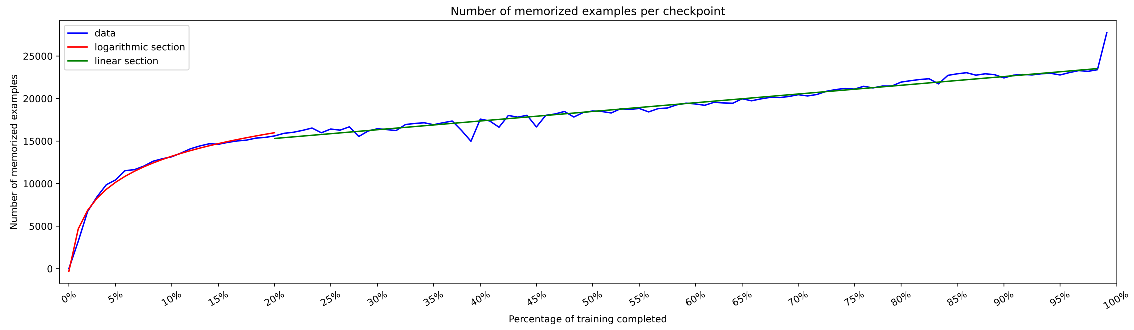

Figure 2: The number of examples memorized by the final checkpoint that are also memorized at each previous checkpoint.

about the nature of memorized examples, our results suggest that sensitive data should be put in the second section, where the memorization rate is the lowest. Since at each checkpoint the model is exposed to a fixed amount of data (22 billion tokens per 5,000 training steps), a higher proportion of data the model is exposed to gets memorized during the first section and last step than during most of the training.

We can further demonstrate this phenomenon by plotting the "memorization delta" at each checkpoint, i.e. the difference between the number of examples memorized at each checkpoint compared to the prior checkpoint. The results are shown in Figure 3.

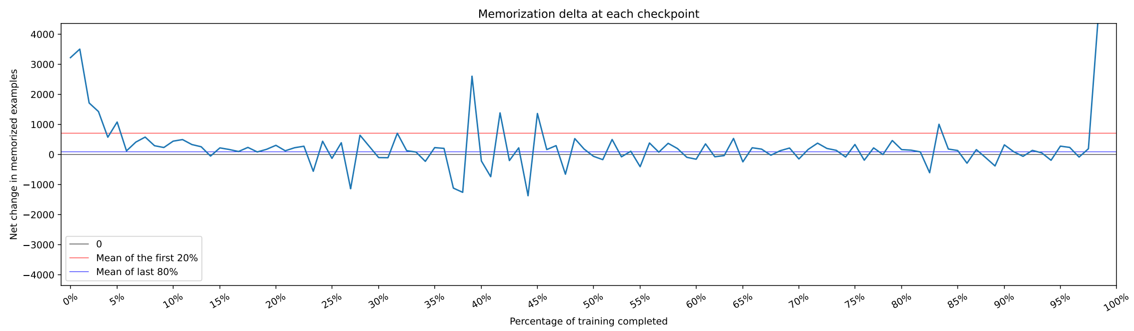

Figure 3: The memorization delta at each checkpoint.

Figure 3 paints a clear picture: the memorization rate decays then stabilizes at a slightly positive value and finally spikes at the last checkpoint. In the first 20%, each checkpoint has an average of 665.86 more examples memorized than the last checkpoint. Then for the last 80% (excluding the last step), the memorization delta is only 86.63 examples on average. But at the final checkpoint, 4,177 more examples are memorized than at the previous checkpoint.

Note that this increased growth does not point to anything special about the last checkpoint. The OLMo authors do not specify that step 555,000 in the training was any different than the previous steps. And indeed, our results in Section 4.3 show that if you filter to only examples memorized at any given checkpoint, it appears that that checkpoint has memorized a disproportionate number of examples. This phenomenon is discussed more in that section.

We go a step further and decompose the memorization delta at each stage into two components: the number of "newly memorized" examples and the number of forgotten examples at each checkpoint compared to the prior checkpoint. We calculate the number of "newly memorized" examples by taking the examples memorized at each checkpoint and checking whether they were memorized at the previous checkpoint as well. Similarly, we calculate the number of forgotten examples by taking the memorized examples at the prior checkpoint and seeing how many of them are not memorized at the current checkpoint. Subtracting the number of examples forgotten by each checkpoint from the number of examples newly memorized by each checkpoint is equivalent to the memorization delta in Figure 3. The results of plotting the newly memorized and forgotten examples at each checkpoint is shown in Figure 4.

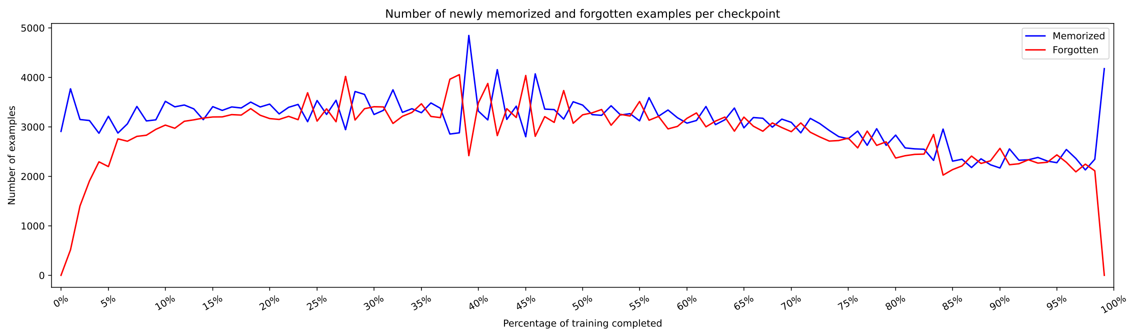

Figure 4: The number of newly memorized and forgotten examples at each checkpoint.

Figure 4 illustrates what causes the decay in memorization rate early on in the training process: it is not that these checkpoints have newly memorized more examples, rather, they have forgotten fewer examples. If the high memorization rate early on was caused by lots of new examples being memorized, we would see the blue line starting high and then decaying to meet the red line. Instead, we see the red line starting low and then growing logarithmically to meet the blue line.

It is also interesting to notice what appear to be symmetries in the new memorizations and forgetting rates: for many of these checkpoints when memorization goes up, forgetting goes down, and vice versa. This is the cause of drops and rebounds (prominent around 40% of the way through training) visible in Figure 2, as well as the drops and spikes visible in Figure 3. More investigation is needed to understand the mechanisms that cause these drops and spikes.

The fact that these trends are symmetrical rather than correlated implies that some checkpoints see a relatively higher rate of forgetting paired with a relatively lower rate of memorization (and vice versa) than their neighbors. The fact that rebounds in total memorization follow drops in leads these researchers to tentatively conclude that temporarily lowered memorization and raised forgetting make room for rapid consolidation of new memorizations.

Note that the definition of "forgetting" used in this figure does not imply that no future checkpoint will re-memorize the example. Indeed, because for this plot we filtered to only include examples that are memorized at the final model, every example that is "forgotten" at a previous checkpoint has definitionally been re-memorized later on, or else it would not be present in this dataset. This implies the phenomenon we mentioned previously: examples are generally memorized early, sometimes forgotten, and often re-memorized later on.

We investigate this phenomenon by plotting when each example that is memorized by the final checkpoint was *first* memorized, as shown in Figure 5.

There is a significant skew in the distribution of checkpoints at which examples are first memorized. Of the 26,423 examples memorized by the final checkpoint, 50% of them were memorized within the first 6% of training. While we can see a small spike in the number of examples first memorized by the last checkpoint, the majority of

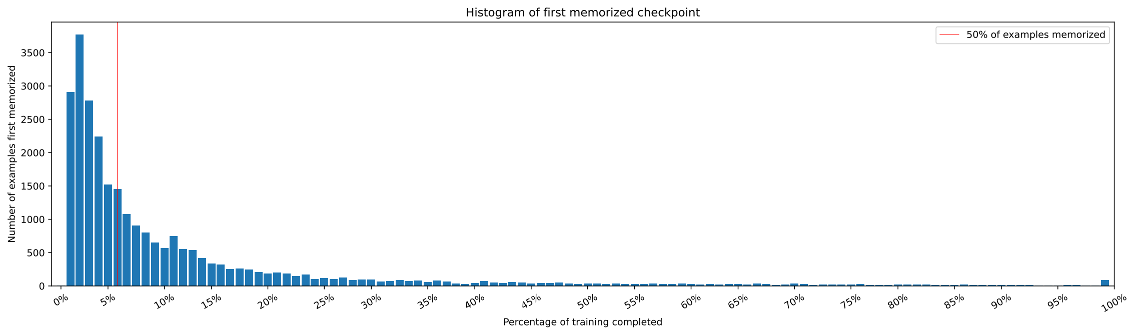

Figure 5: The number of examples that are memorized for the first time at each checkpoint.

examples that are memorized by the final model are actually first memorized very early in the training process. This is different than the behavior observed in [20], which found that, in computer vision models, memorization occurs more frequently in later parts of the training process.

The previous plot shows that a majority of examples are first memorized early in training but we know that many of these examples will be forgotten throughout the training process and then re-memorized later. To understand the relation between these phenomena, we also create a plot that shows the start of memorization "streaks" which terminate at the final checkpoint. We define a memorization "streak" for an example as a set of contiguous checkpoints, all of which have memorized that example. To find the beginnings of streaks that end at the final checkpoint, we take all of the examples memorized by the final checkpoint and then work backward, seeing at which checkpoint each example was first memorized within that streak. We then plot the distribution of these streak-start checkpoints, as shown in Figure 6.

Figure 6 is almost a mirror image of the prior plot: while there are 1,127 examples that are memorized continuously throughout the entire training run, the vast majority of examples learned early are forgotten and then re-memorized later on. More than 50% of final streaks are started after 90% of training is complete, and the final checkpoint alone accounts for more than 15% of the 26,423 examples memorized.

Combining the insights from these two visualizations, we characterize the memo-

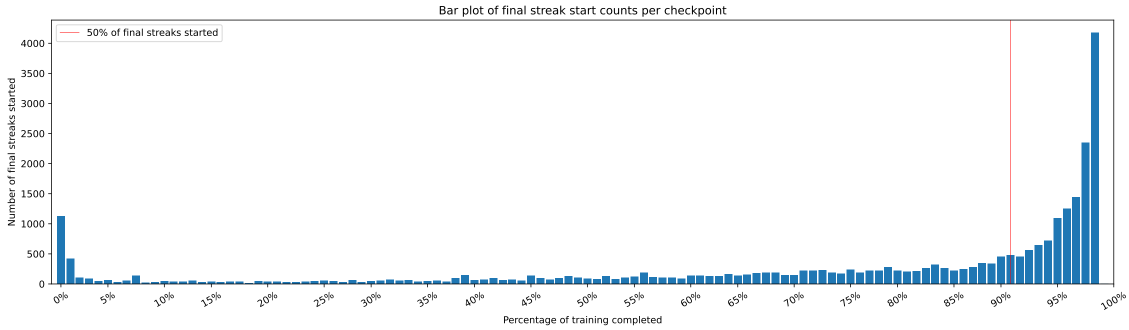

Figure 6: The number of "final streaks" that are started at each checkpoint.

rization behavior of models as such: models memorize a great deal of the training data they are initially exposed to, then forget much of it, then re-memorize some of it. It's worth noting that, though it might appear that models re-memorize a great number of examples close to the end of training, this is actually a statistical artifact: since we are only showing examples memorized by the final checkpoint, there is a bias towards "final streaks" starting near the final checkpoint. This motivates our work in Section 4.3.

4.3 Intermediate Checkpoint Analysis

Although the previous analysis focused on the "final" checkpoint, it is important to note that the choice of when to end training is somewhat arbitrary. Although heuristics like the Chinchilla scaling laws [32] provide guidance for the compute-optimal amount to train a model, researchers often decide when to stop training based on compute or training data constraints. As such, intermediate checkpoints can be equally useful to analyze. In fact, they provide an opportunity to study an interesting counterfactual scenario: what would have happened to the examples memorized by the "final" checkpoint if researchers had continued to train the model?

We arbitrarily select the checkpoint by which 75% training has been completed and filter to only select examples that are memorized at our intermediate model. Reproducing Figure 2, we plot how many of these examples are memorized at each checkpoint in Figure 7.

This plot follows a similar structure to Figure 2, with the same curving growth

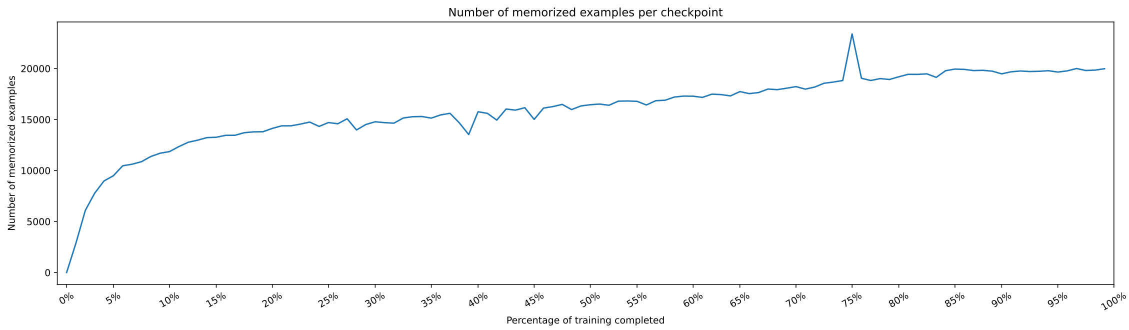

Figure 7: The number of examples memorized by an intermediate checkpoint that are memorized at each checkpoint.

turning into linear growth. The difference is that, rather than having a spike at the last checkpoint, there is instead a spike and immediate drop which correspond to the checkpoint we are analyzing. This indicates that many of the examples memorized at our checkpoint are memorized *by* that checkpoint (4,722 or 19.12% of the total number of examples memorized at checkpoint 75%) but that many examples are also forgotten at the next stage (4,538 or 18.38%).

Table 2: New memorizations at 75% and forgetting at 75%+1.

Memorized at step 75%	**Newly memorized at step 75%**	**Forgotten at step 75%+1**
24,687	4,722 (19.12%)	4,538 (18.38%)

This raises an interesting question: are the examples forgotten at step 75%+1 primarily examples that the model has just learned at step 75%, or are they examples that the model learned earlier in training? We decompose the previous and future states of examples memorized at 75% in table 3.

The vast majority of examples (17,740 or 71.86%) memorized at checkpoint 75% were also memorized at step 75%-1 and remained memorized at step 75%+1. Of examples that were newly memorized at 75%, about half remained memorized at 75%+1 (2,409, 51.02%) and half were forgotten at step 75%+1 (2,313, 48.98%). Similarly, of examples that were forgotten at 75%+1, about half were newly memorized at step 75% (2,313, 50.97%) and about half had also been memorized at step 75%-1 (2,225, 49.03%).

Table 3: Previous and future states of examples memorized at 75%.

	Newly memorized at step 75%	**Memorized at step 75%-1**	**Total**
Remain memorized at 75%+1	2,409	17,740	20,149
Forgotten at 75%+1	2,313	2,225	4,538
Total	4,722	19,965	24,687

Another underlying trend we can analyze by looking at the intermediate checkpoint is the novelty of memorization and the permanence of forgetting. Of the 4,722 examples that were newly memorized at step 75%, only 129 (2.73%) had never been memorized before. Of the 4,538 examples memorized at 75% that are forgotten at 75%+1, only 102 (2.25%) were never memorized again. This all reinforces a key insight of this work: most examples are memorized early, then periodically forgotten and re-memorized throughout the training process.

4.4 Initial Checkpoint Analysis

To further understand how the examples memorized early on are forgotten and re-memorized throughout training, we examine the examples memorized by an early checkpoint to see how they fare. The very first checkpoint has no memorized examples because it has not been exposed to any training data, so we select the checkpoint after that to better understand the memorization dynamics early in training and see which of those examples remain memorized throughout training, as shown in Figure 8

By plotting the number of examples that were memorized at our initial checkpoint which are also memorized at other checkpoints, we can see a very strong and simple trend: in the first steps of training, 3,096 examples are memorized, and over the course of training, few are forgotten. Notably, very few of these memorized examples are forgotten at the final checkpoint: only 188 (6.07%) of the examples memorized

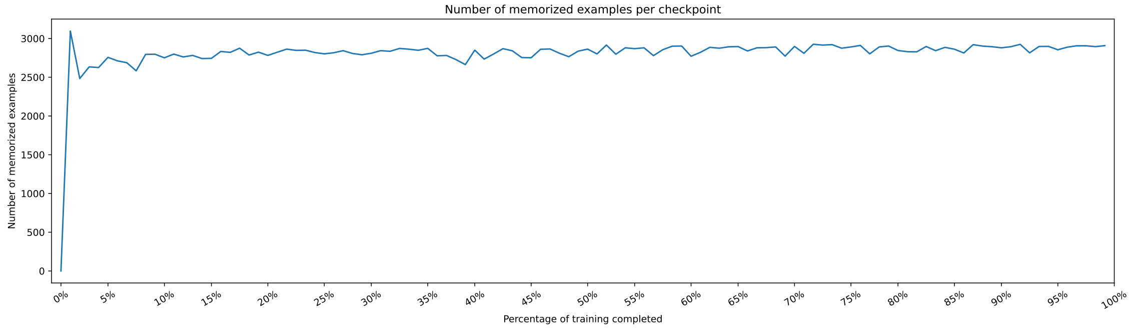

Figure 8: The number of examples memorized by the initial checkpoint that are memorized at each checkpoint.

at the initial checkpoint. This implies that examples memorized early on crystalize in the LLM's representation and are unlikely to be forgotten.

We also illustrate this diminishing crystallization by taking the examples memorized at each of the first 10 checkpoints and calculating what proportion of them are continuously memorized for the last 80% of training. We take this ratio to be indicative of the percentage of memorized examples at each checkpoint that are "crystalized" and remain memorized throughout much of training, and plot the results in Figure 9.

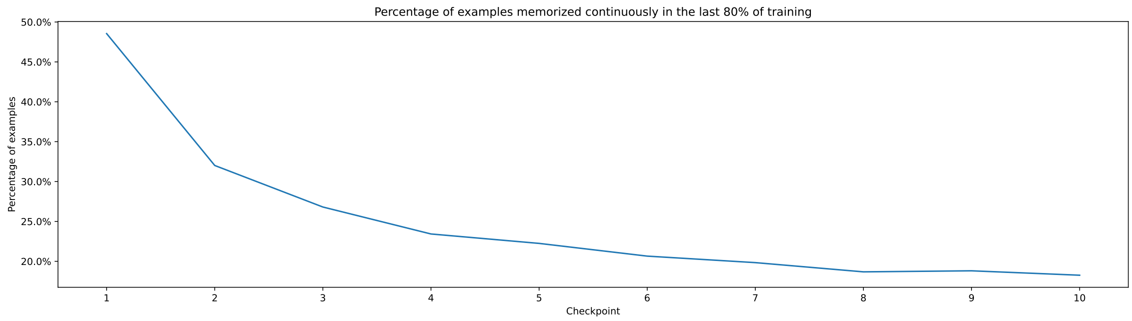

Figure 9: The percentage of examples memorized at each checkpoint that are memorized continuously in the last 80% of training.

Of the 3,096 examples memorized by the first checkpoint, 1,503 (48.55%) are memorized continuously for the last 80% of training. For each subsequent checkpoint, this percentage decays logarithmically, until it reaches a stable forgetting rate at around 20% of examples memorized.

All of this analysis illustrates that, while many examples are forgotten and re-memorized throughout training, the examples memorized early on are most likely to stay crystalized throughout all of training, while examples memorized later are less likely to be crystallized. This points to the powerful impact of training order on memorization rate.

4.5 Classification

Table 4: Percentages of random and memorized data which is automatically classified into each category. A Z-score is provided to quantify the extent to which the percentages differ and in which direction.

Classification	**Random**	**Memorized**	**Z-score**
License, terms of use, copyright notices	1.38%	6.95%	76.995659
Promotional content (products, subscriptions, etc.)	**32.68%**	**51.91%**	54.689442
Contact info (address, email, phone, twitter, etc.)	0.48%	1.59%	26.280689
Log files and error reports	0.15%	0.75%	25.300248
Code	7.15%	**11.05%**	23.773840
Configuration files	0.64%	1.72%	22.065379
Pseudonyms	0.16%	0.57%	17.179114
Web forms (menu items, instructions, etc.)	1.98%	3.40%	16.500842
Lists of numbers (dates, sequences, etc.)	0.80%	1.32%	9.569155
Valid URLs	0.03%	0.08%	5.291503
Lists of named items (games, countries, etc.)	8.30%	**8.85%**	3.117694
High entropy (UUIDs, base64 data)	0.22%	0.20%	-0.662266
Donald Trump tweets and quotes	0.08%	0.02%	-3.837613
Other/misc	0.95%	0.14%	-13.534548
Religious texts	2.35%	0.86%	-15.770547
Tech news	9.95%	2.41%	-38.818647
Named individuals (non-news samples only)	**10.59%**	2.58%	-39.972264
Forum or Wiki entry	**15.00%**	5.36%	-40.454061
US and international news	7.15%	0.22%	-42.105140

As shown in Table 4, the three most common categories of random samples were promotional content (32.68%), forum or wiki entries (15%), and named individuals (10.59%), while the three most common categories of examples memorized by the final checkpoint were promotional content (51.91%), code (11.05%), and lists of named items (8.85%). Performing a chi-squared test, the two samples are statistically significantly different, with a p-value approaching 0. This implies that there is a relationship between the classification category and whether a sample is memorized or not. We can further visually confirm this by comparing the two groups in Figure 10.

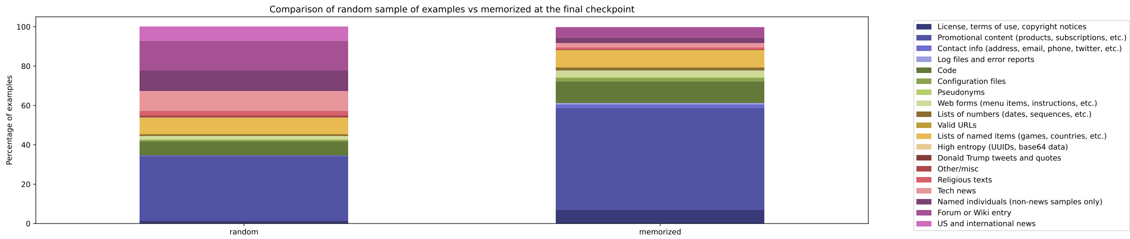

Figure 10: A comparison of the examples from a random sample vs the examples memorized at the final checkpoint.

Understanding that the distributions of categories are different between the two groups prompted us to ask: if category really is correlated with memorization, is it also correlated with the patterns of memorization we've observed throughout training? Do the memorization dynamics change according to the category of example being investigated? To investigate these, we plotted the number of memorized examples per checkpoint by category. Because the categories have different magnitudes, we used a logarithmic plot to be able to see the trends for all categories, as shown in Figure 11.

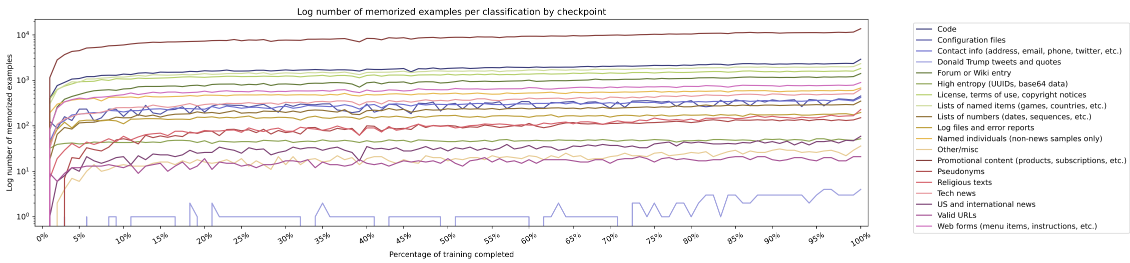

Figure 11: The number of examples memorized per checkpoint by category, plotted on a logarithmic y-axis.

The most commonly memorized categories of example seemed to follow similar memorization dynamics as observed for the dataset as a whole, though some of the infrequently memorized categories have too few samples to be able to conclude in their case. This implies that, given no other evidence, the memorization and forgetting dynamics are the same for all examples, regardless of which category they're in.

We were interested in using category as a proxy for how sensitive data may be. Our analysis focuses on memorization without any particular preference for memorization of content that may be considered private, personal, or whose memorization results in disproportionate harm. However, as discussed in the Introduction, one goal of memorization research is to minimize the extent of memorization of sensitive data. When we examine the category that most obviously represents sensitive data ("Contact info (address, email, phone, twitter, etc.)"), we see in Figure 12 that it largely follows the same trends and patterns as the overall dataset.

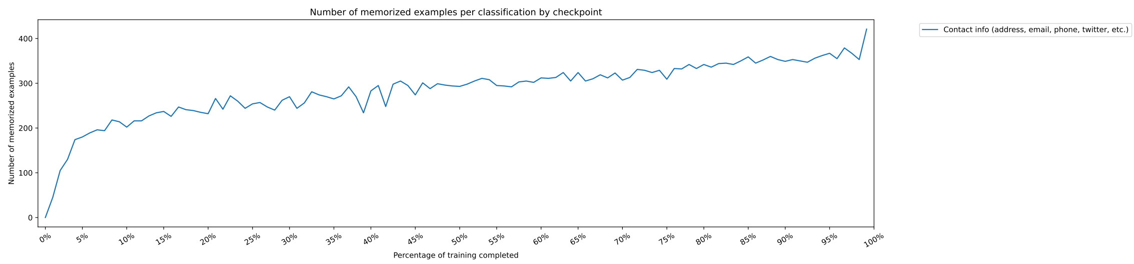

Figure 12: The number of examples categories as "contact info" that are memorized at each checkpoint.

This category appears three times as often in the memorized dataset as in the random sample dataset, which confirms previous research [33] that this type of sensitive data can indeed get memorized.

We were also curious to understand how the examples memorized at every checkpoint in the last 80% of training differed from the ones memorized at the last checkpoint. These examples are memorized continuously at many checkpoints, implying that they are the most crystalized. We hypothesized that they would have a different distribution than the ones memorized at the end, which include some examples which hadn't been memorized before the final checkpoint.

As shown in Table 5 and Figure 13, not only are examples labeled as "Contact info" more likely to be memorized, they are more likely to stay memorized throughout training. To explore this, in a qualitative analysis of the examples memorized in the

Table 5: Percentages of memorized at the final model and memorized in all of the last 80% of training which are automatically classified into each category. A Z-score is provided to quantify the extent to which the percentages differ and in which direction.

Classification	**Memorized at final**	**Memorized at all last 80%**	**Z-score**
Named individuals (non-news samples only)	2.58%	8.07%	17
High entropy (UUIDs, base64 data)	0.20%	1.19%	11
License, terms of use, copyright notices	6.95%	11.26%	8
Code	**11.05%**	**15.23%**	6
Log files and error reports	0.75%	1.75%	6
Lists of named items (games, countries, etc.)	**8.85%**	**12.17%**	5
Valid URLs	0.08%	0.32%	4
Contact info (address, email, phone, twitter, etc.)	1.59%	2.23%	2
Other/misc	0.14%	0.04%	-1
Web forms (menu items, instructions, etc.)	3.40%	2.90%	-1
Lists of numbers (dates, sequences, etc.)	1.32%	0.99%	-1
US and international news	0.22%	0.04%	-2
Configuration files	1.72%	0.80%	-4
Religious texts	0.86%	0.16%	-4
Tech news	2.41%	1.03%	-4
Forum or Wiki entry	5.36%	2.03%	-7
Promotional content (products, subscriptions, etc.)	**51.91%**	**39.78%**	-9
Pseudonyms	0.57%	0.00%	-999
Donald Trump tweets and quotes	0.02%	0.00%	-999

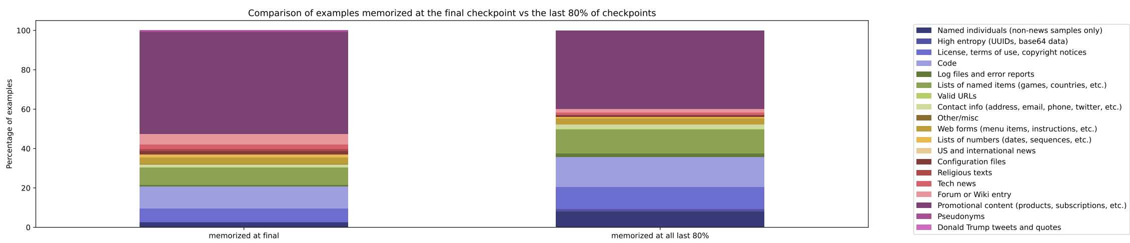

Figure 13: A comparison of the examples memorized at the final checkpoint vs the last 80% of checkpoints.

last 80%, we found a strong pattern that has not been described in previous research on memorization. Examples that are memorized in the last 80% are more likely to follow a "templated" structure (examples in Section 7.5). These templated examples are essentially repeated, with only minor and consistent variation, allowing them to be easily memorized by the Attention mechanism. However, since they are not identical duplicates, they do not get removed by the deduplication techniques used in this paper or in OLMo's training data [26].

Some examples of templates:

> Please find here the contact details and reviews of [NAME] in [CITY]. Info on [NAME] includes the main phone number for [NAME], a fax number, plus the street address for [NAME] in [CITY] with clean directions to

> opening times, address and phone number. [JOB CENTER] Jobcentre to find out about the benefits and support you're entitled to including; Tax Credits, Universal Credit, Income Support, Jobseekers Allowance, ESA and Pension Credit.\n[JOB CENTER] doesn't (JOB CENTER)

> Where is [BUSINESS] located?\n[BUSINESS] is located at: [ADDRESS].\nWhat is [BUSINESS]'s phone number?\n[BUSINESS]'s phone number is: [PHONE NUMBER]

> How can I contact [NAME]'s management team or agent details, and how do I get in touch directly?\nAre you looking to get in touch with [NAME] for commercial opportunities? Key contact information and audience insights, can be found on The Handbook, an online resource for accessing influ

Many contain personal information such as names, phone numbers, addresses, email addresses, etc. This explains why the examples memorized in the last 80% have an even higher proportion of examples classified as "Contact info" than examples memorized at the final model. Personal information embedded in templates can be more easily identified and removed as part of the training data preparation process. The curators of OLMo's training dataset filtered data automatically identified as personally identifiable information [26], but examples such as the ones in Section 7.5 were still present in the training data and did still end up memorized by the model.

Further work needs to be done to compare the distributions of examples that are newly memorized at each checkpoint but these authors suspect that, at early checkpoints, the models memorize examples at a ratio more similar to the random sample.

Chapter 5

Limitations

5.1 High Sensitivity

As discussed in Section 2.1, the method of extracting memorized sequences used in this research is not representative of realistic membership inference attacks. By both prompting the models with exact samples from their training data and using greedy decoding, we maximize the probability that a memorized example will be output. In the real world, attackers are unlikely to have access to the training data and therefore are unlikely to be able to feed it verbatim to the LLM. Additionally, if they did have access to the training data, there would be no purpose in attempting to extract training data from the model. Another factor that contributes to the unrealisticness of this method of attack is that most commercially available LLM providers do not use greedy decoding since it produces highly-repetitive text [34].

Although this attack method is unrealistic, we think this area of research is still useful because it allows us to understand all information that is potentially memorized by the model. Since the two things that make this method unrealistic (prompting with exact training data and greedy decoding) also make the model more likely to produce any data it may have memorized, we view our approach as highly sensitive, extracting a large portion of all memorized data, and therefore acting as a canary in the coal mine.

5.2 Lack of Randomized Experimentation

Ultimately, some of our conclusions and tentative recommendations can only be fully confirmed by running randomized experiments. For example, although we infer that model developers should put sensitive training data early but not too early in the training process, it is possible that there are confounding effects that would actually cause this data to be memorized at the same rate, regardless of where it was put. We lack the resources to experiment with training orders but hope to motivate future researchers to test these recommendations more robustly.

5.3 Potential Misclassification

In a qualitative analysis, we found mixed results from the quality of our automated classification process. As discussed in Subsection 7.4.2, some categories saw high rates of misclassification, while in other cases the automated classifications matched the authors' classifications. This is partially attributable to the fact that some 64 token strings are inherently challenging to classify since many do not contain enough information to be confidently classified, even by the authors. However, some of the examples were clearly misclassified, potentially degrading the quality of results of the classification analysis. We leave improved classification analysis for future work.

5.4 Only English-Language Analysis

As OLMo is a model primarily trained on English text data [26] and intended for use in English [25], very few of the memorized examples we encountered in manual analyses were in languages other than English. There are documented attack vectors which take advantage of low-resource languages to bypass LLM safeguards [35] and it is possible that there are ways to extract training data from LLMs by using low-resource languages. It is also possible that different languages have different memorization dynamics, so further research needs to be performed to understand whether the phenomena we describe are limited to English or would apply to other languages as well.

Chapter 6

Conclusion

Memorization in LLMs is a well-documented phenomenon but more work needs to be done to understand how that memorization occurs, what data is most likely to be memorized, and what can be done to minimize undesirable memorization. This field of research is important for making LLMs useful in commercial applications, as memorization can result in leaking of private information and copyright infringement.

We have made novel contributions by exploring previously unresearched dynamics of memorization throughout the training process. By analyzing memorization at various checkpoints along the training of an LLM, we are able to come to some important conclusions. Most significantly:

1. Models memorize more data earlier on in training
2. Models forget examples during the training process
3. Many forgotten examples are re-memorized during training
4. Different types of text get memorized at different rates, but the dynamics of memorization across training are not necessarily different

However, much work remains to be done in this field. Some areas of exploration that we think could be fruitful for future research:

1. Controlled experimentation on how training order might influence memorization
2. How the phenomenon of "templated" examples in training might be used to further deduplicate data and decrease memorization
3. More rigorous classifications processes to analyze how different types of data get memorized
4. Research into memorization in languages other than English

We hope that future researchers find this work useful in understanding the impact of training order on memorization and forgetting in LLMs.

Chapter 7

Appendix

7.1 Soft Memorization Metrics

All prior work on memorization that we surveyed used a strict definition of extractability, i.e. checking whether the generated output exactly matched the continuation of the sequence in the training data. This is a convenient metric to use because it is reliably easy to evaluate without human intervention. However, for both the goals of evaluating undesirable semantic memorization and mitigating copyright violations, this is an overly strict definition of "memorization". Therefore, rather than only evaluate extractability according to the "hard" definition (previously defined in the Section 2.1), we also propose a "soft" definition of intractability: A string s is ***d**-extractable with k tokens of context* from a model f if there exists a (length-k) string p, such that the concatenation $[p \parallel s]$ is contained in the training data for f, and f produces a string which is at most distance d away from s when prompted with p using greedy decoding, for some specified distance measure.

To evaluate whether relaxing the definition of memorization by using d-extractability changed our observed memorization dynamics, we calculated Hamming distance and Levenshtein distance as well as the longest common subsequence for each of the generations in the training dataset.

1. Hamming distance: the number of characters that need to be changed in place to make two equal-length sequences identical [36]

2. Levenshtein distance: the number of characters that need to be inserted, deleted, or substituted to make two sequences identical [37]

3. Longest common subsequence similarity: the length of the longest continuous sequence of characters that two sequences have in common [38]

We used the Python package “textdistance” [39] to efficiently evaluate these similarity metrics.

Performing the same analysis that we had done previously required discretizing these continuous distance metrics, i.e. selecting a value for d. We decided to select these values based on the distribution of each distance metric and arbitrarily selected 2.5%, 5%, 10%, 15%, and 25% quantiles for this cutoff. For example, the 5% quantile represents a cutoff which will treat 5% of the examples as memorized. We reproduced Figure 2 for all three distance measures and all five quantiles in Figure 14.

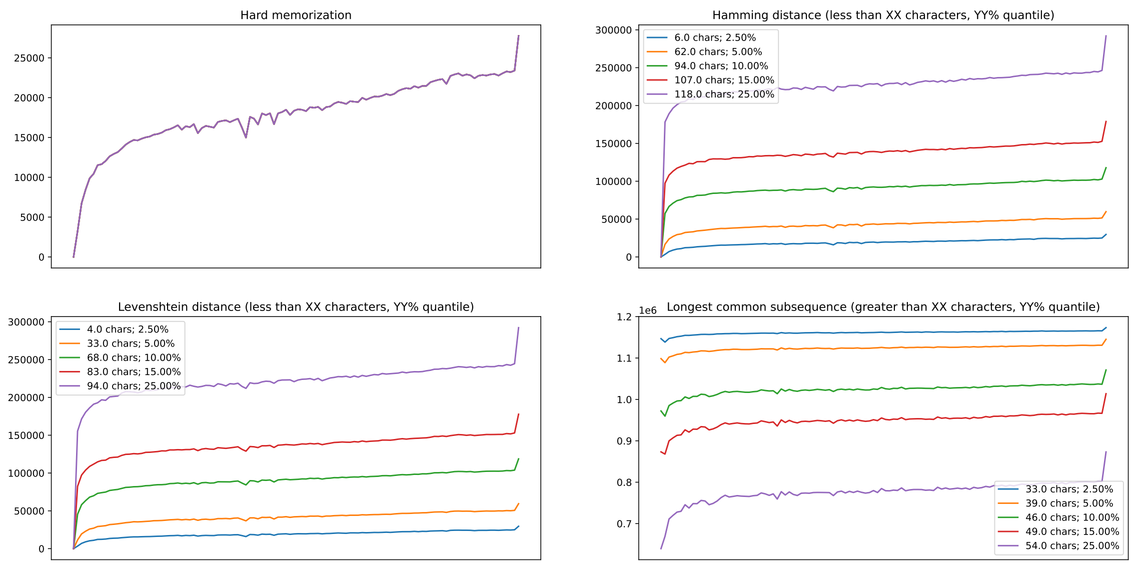

Figure 14: The number of examples that qualify as "memorized" at each checkpoint, using a variety of distance measures and cutoffs.

For all three similarity measures we evaluated, and for all five quantiles, the shapes of the graphs were not meaningfully different than the shape we saw when using the hard definition of memorization: a logarithmic increase followed by a linear increase. When we experimented with different cutoffs, we found that the same shape was generally preserved, except for cutoffs that represented extreme relaxations of the memorization criterion.

We wanted to further investigate whether a different cutoff could help us better understand the memorization dynamics. To do this, we calculated the cutoff values for all 0.01% increments of the cutoff quantiles, and plotted the cutoff values against the percentage of examples that would be treated as “memorized” if we used that cutoff. The results are in Figure 15.

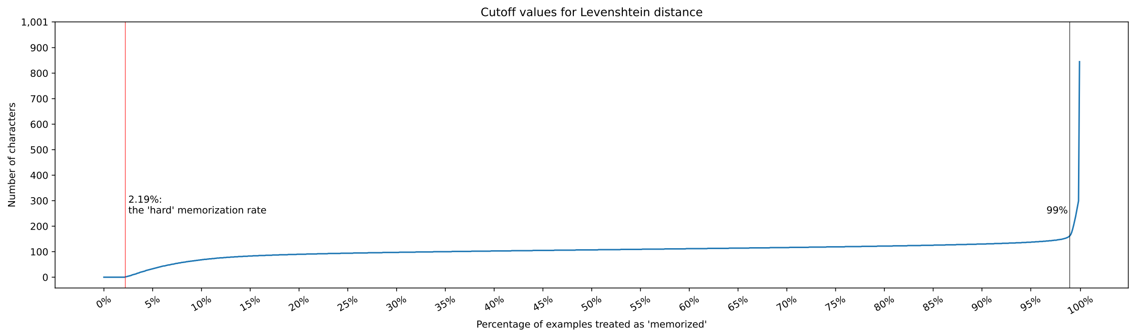

Figure 15: For a percentage of examples to be treated as "memorized", this plots the corresponding number of characters that would be used for a cutoff for Levenshtein distance.

The lack of meaningful inflection points in the graph indicates that these metrics are best understood as continuous measures, rather than being discretized. The first inflection point happens at 2.19%, at which point the cutoff is greater than 0. At a cutoff of 0, the soft memorization metric is equivalent to the hard memorization metric, because the generated text has 0 distance from the expected text. Therefore, this inflection happens at 2.19% because that is the memorization rate according to the hard definition. The other inflection point happens at the 99% mark, which we do not find relevant to this analysis because we do not consider a memorization rate of 99% to be meaningful. Lacking meaningful inflection points indicates to us that there is no trivial and meaningful definition of memorization.

A limitation of all of the metrics we examined is that they do not capture the semantic content of the generations, only making character-wise comparisons. In future work, we hope to further explore meaningful relationships between the definition of memorization used and the trends observed in memorization and forgetting phenomena.

7.2 Pythia

OLMo models are trained using the frequently-used policy of learning rate warmup and annealing, in which the learning rate of an LLM is changed throughout the training process. Specifically, the learning rate is warmed up over the first 5,000 steps and then decayed linearly from there to a tenth of the peak of the learning rate throughout the rest of training. Since [17] showed that learning rate impacts memorization, we were curious to what extent our results could be explained by changes in the learning rate throughout the training process. As a result, we reproduced our work in OLMo with a similarly-sized Pythia [9] model, which has no learning rate warmup or annealing. We found no significant differences to the trends we described with OLMo.

There are some notable differences between the original OLMo work and the Pythia reproduction, namely:

1. How we sampled the Pythia training dataset (described in the Methodology subsection)
2. The amount of duplication (described in the Duplication analysis section)
3. The percentage of memorizations (described in the Results subsection)
4. The classification of different memorized examples (described in the Results subsection)

The fact that, despite all of these differences, the results remained largely the same is heartening evidence that our results generalize to other models.

7.2.1 Framework

Pythia models are trained on The Pile [40] and, like OLMo, release not only final model weights and the training dataset, but also training methodology and checkpoints. Interestingly, the Pythia models also release their training data in the format and order that it is fed to the model during training, which is not information we were able to find on the OLMo model. As a result, we use a different sampling strategy (as described in Subsection 7.2.2) to extract samples for evaluation. We select the 6.9 billion parameter version of Pythia, since it is the most similar in size to the OLMo model we used.

7.2.2 Methodology

Since the authors of Pythia provide the shuffled version of the dataset that they used to train the model, we sampled 1,041,873 examples from evenly-spaced, randomly selected points within the training run, thus ensuring that we would select representative training data. Specifically, we divided Pythia's pre-shuffled Pile dataset's 131,170 iterations into 100 approximately even segments and then selected 10,500 random 64 token sequences from within each segment. We then removed any examples from the same iteration that overlapped as a result of having starts within 64 tokens from each other, resulting in our total of 1,041,873 examples. We then split the 64-token sequences in half, as with OLMo.

The Pythia model has 144 checkpoints separated by 1,000 training steps, starting from step0 and terminating at step143000. We used all of these checkpoints. There are also log-spaced checkpoints provided between step0 and step1000 but we chose not to incorporate these since we wanted evenly-spaced checkpoints.

7.2.3 Results

On the whole, we find our results with Pythia to be nearly identical to our results with OLMo, taking into account some differences caused by sampling noise. Notably, the trends we see in OLMo tend to be less pronounced but still present for Pythia.

We have included Pythia reproductions of all of the major figures and tables we used for our OLMo analysis without comment.

Table 6: Descriptive statistics about our evaluation of memorization in pythia.

Sequences evaluated	**Checkpoints used**	**Memorized by at least one checkpoint**	**Memorized by the final checkpoint**	**Memorized at every checkpoint**
1,027,756	114	10,236 (1% of total sequences)	4,877 (0.47% of total)	98 (0.01% of total)

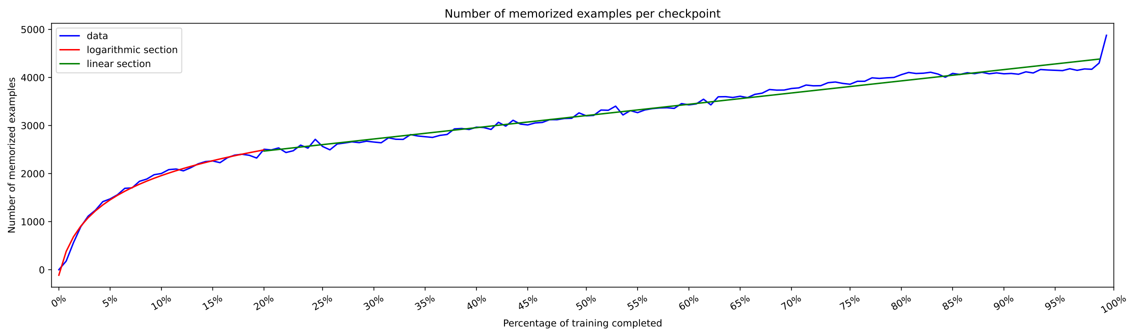

Figure 16: The number of examples memorized by the final checkpoint that are also memorized at each previous checkpoint.

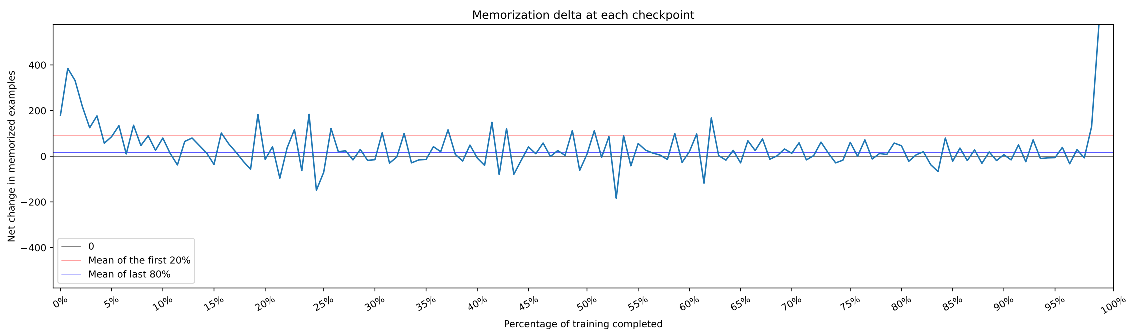

Figure 17: The memorization delta at each checkpoint.

Table 7: New memorizations at 75% and forgetting at 75%+1.

Memorized at step 75%	**Newly memorized at step 75%**	**Forgotten at step 75%+1**
4,614	689 (14.9%)	660 (14.3%)

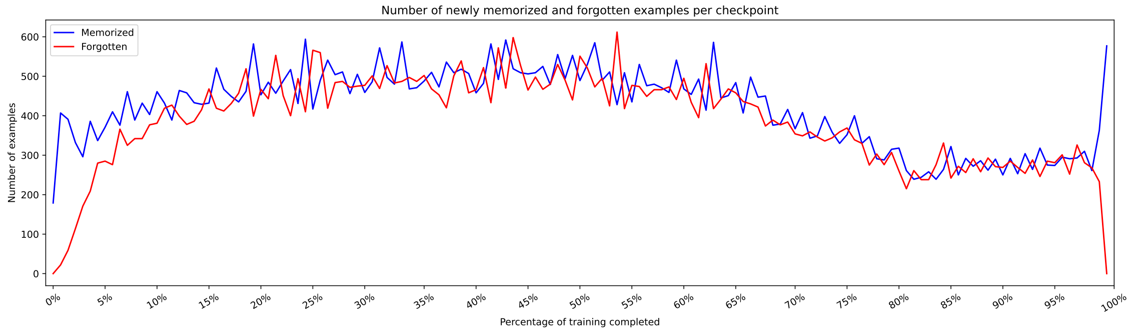

Figure 18: The number of newly memorized and forgotten examples at each checkpoint.

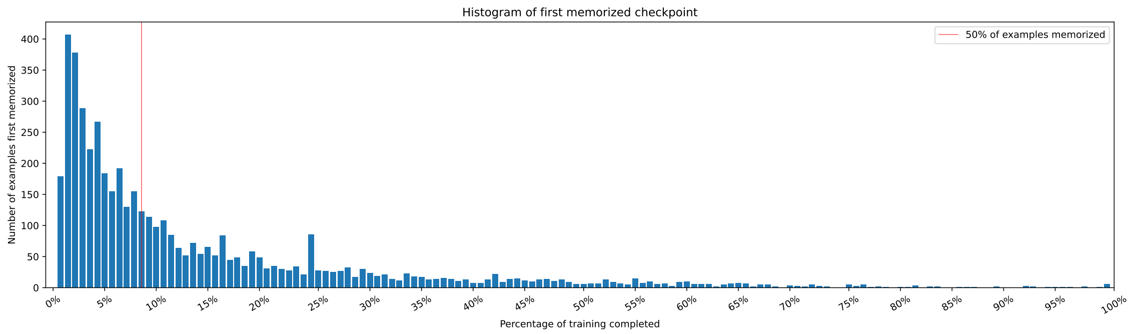

Figure 19: The number of examples that are memorized for the first time at each checkpoint.

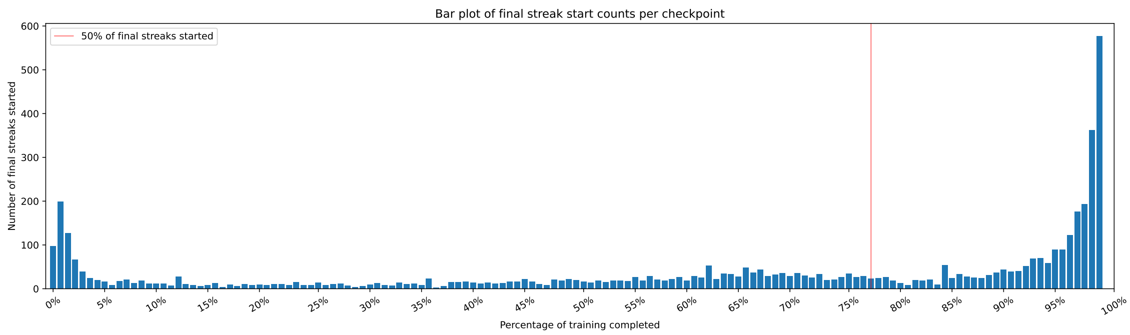

Figure 20: The number of "final streaks" that are started at each checkpoint.

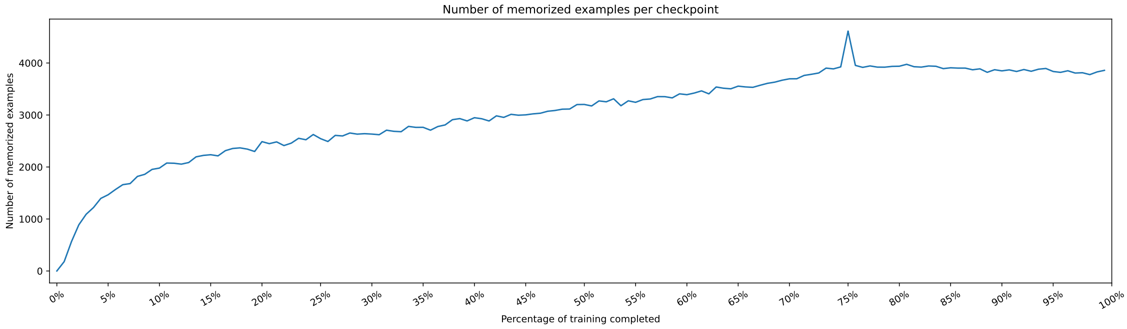

Figure 21: The number of examples memorized by an intermediate checkpoint that are memorized at each checkpoint.

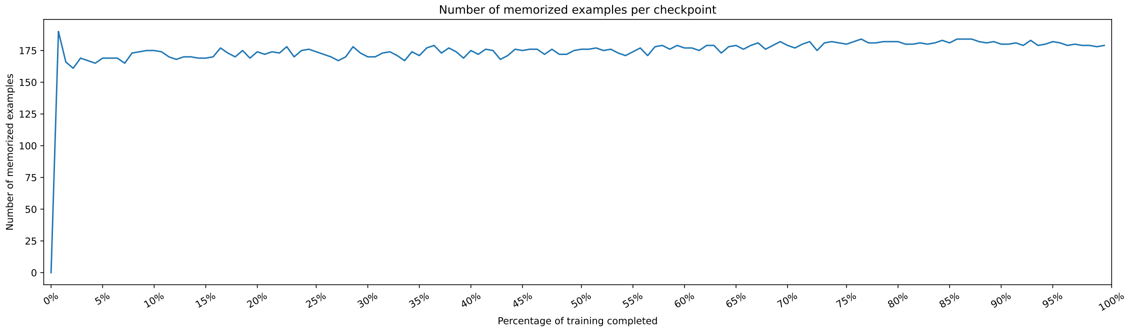

Figure 22: The number of examples memorized by the initial checkpoint that are memorized at each checkpoint.

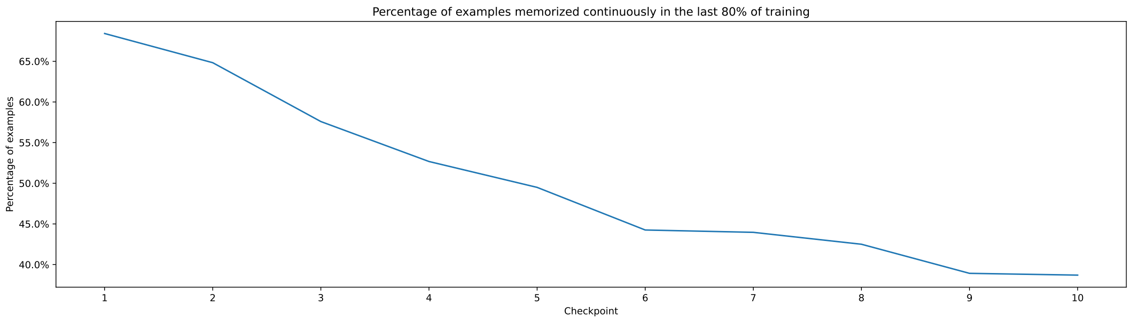

Figure 23: The percentage of examples memorized at each checkpoint that are memorized continuously in the last 80% of training.

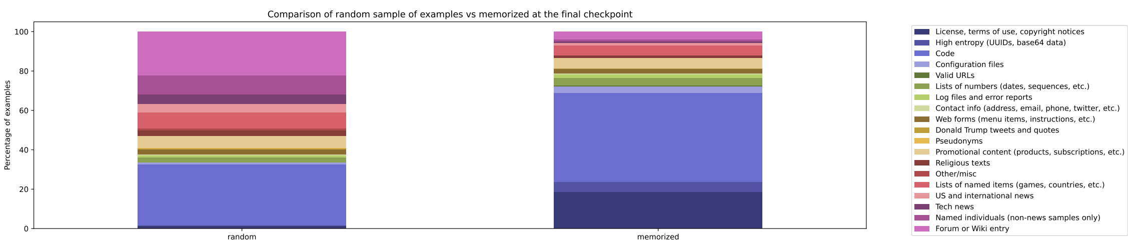

Figure 24: A comparison of the examples from a random sample vs the examples memorized at the final checkpoint.

Table 8: Percentages of random and memorized data which is automatically classified into each category. A Z-score is provided to quantify the extent to which the percentages differ and in which direction.

Classification	**Random**	**Memorized**	**Z-score**
License, terms of use, copyright notices	1.23%	18.68%	109.863628
High entropy (UUIDs, base64 data)	0.45%	5.13%	48.609763
Code	30.92%	45.07%	17.768404
Configuration files	0.94%	3.40%	17.693035
Valid URLs	0.06%	0.31%	6.928203
Lists of numbers (dates, sequences, etc.)	2.58%	4.00%	6.147009
Log files and error reports	1.05%	1.66%	4.200840
Contact info (address, email, phone, twitter, etc.)	0.45%	0.57%	1.279204
Web forms (menu items, instructions, etc.)	2.65%	2.32%	-1.408721
Donald Trump tweets and quotes	0.29%	0.14%	-1.870829
Pseudonyms	0.16%	0.02%	-2.474874
Promotional content (products, subscriptions, etc.)	6.34%	5.35%	-2.730625
Religious texts	2.79%	1.35%	-6.002450
Other/misc	1.03%	0.08%	-6.505382
Lists of named items (games, countries, etc.)	8.04%	4.94%	-7.626652
US and international news	4.16%	1.33%	-9.685701
Tech news	5.06%	0.84%	-13.107466
Named individuals (non-news samples only)	9.49%	0.82%	-19.658477
Forum or Wiki entry	22.31%	3.98%	-27.103356

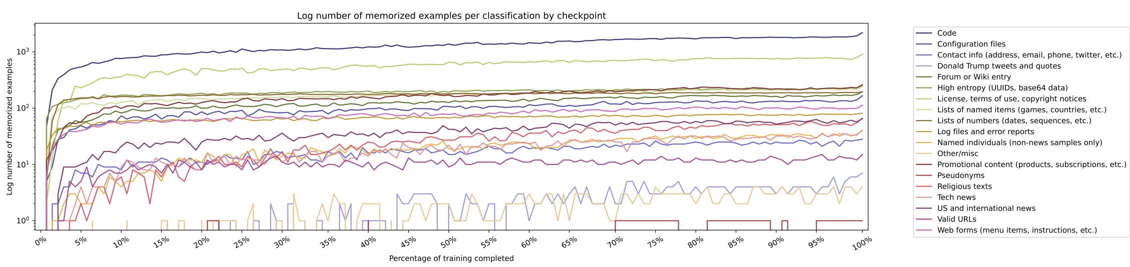

Figure 25: The number of examples memorized per checkpoint by category, plotted on a logarithmic y-axis.

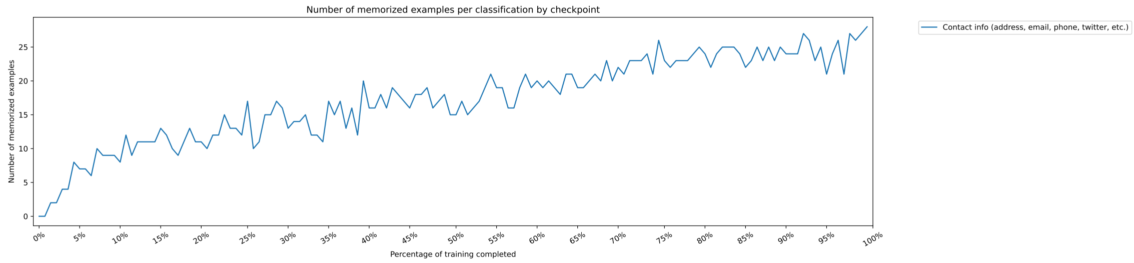

Figure 26: The number of examples categories as "contact info" that are memorized at each checkpoint.

Table 9: Percentages of memorized at the final model and memorized in all of the last 80% of training which are automatically classified into each category. A Z-score is provided to quantify the extent to which the percentages differ and in which direction.

Classification	**Memorized at final**	**Memorized at all last 80%**	**Z-score**
Lists of numbers (dates, sequences, etc.)	4.00%	17.73%	20
High entropy (UUIDs, base64 data)	5.13%	12.42%	9
Lists of named items (games, countries, etc.)	4.94%	10.37%	7
Log files and error reports	1.66%	4.46%	6
Web forms (menu items, instructions, etc.)	2.32%	3.38%	2
Forum or Wiki entry	3.98%	3.86%	-0
Valid URLs	0.31%	0.24%	-0
Contact info (address, email, phone, twitter, etc.)	0.57%	0.36%	-1
US and international news	1.33%	0.60%	-2
Configuration files	3.40%	2.05%	-2
Promotional content (products, subscriptions, etc.)	5.35%	3.62%	-2
Named individuals (non-news samples only)	0.82%	0.12%	-2
Tech news	0.84%	0.12%	-2
Code	45.07%	36.19%	-4
License, terms of use, copyright notices	18.68%	4.46%	-10
Religious texts	1.35%	0.00%	-999
Donald Trump tweets and quotes	0.14%	0.00%	-999
Other/misc	0.08%	0.00%	-999
Pseudonyms	0.02%	0.00%	-999

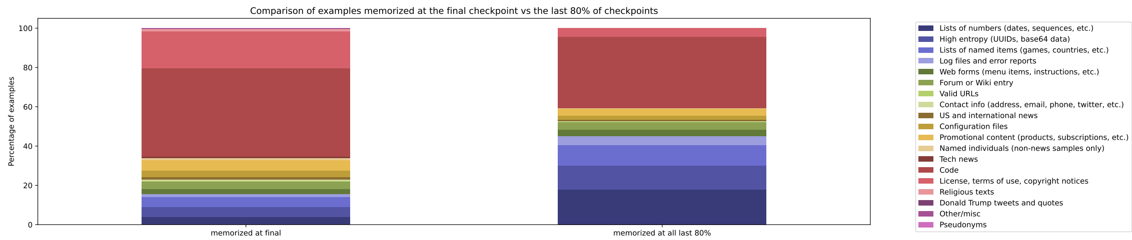

Figure 27: A comparison of the examples memorized at the final checkpoint vs the last 80% of checkpoints.

7.3 Duplication Analysis

Prior research [16] has shown that repeated examples in the training data are more likely to be memorized but that deduplication does not entirely prevent memorization. Although both the OLMo and Pythia models are trained on datasets that have been heavily deduplicated, some sequences repeat in various places in the training data. To understand whether the phenomena we observe in our analysis result from duplicated data, we analyzed duplication in our sequence set for both OLMo and Pythia.

Model	**Duplicated sequences in overall dataset**	**Duplicated sequences memorized by the final checkpoint**
OLMo	2,428 (0.2%)	1,419 (4.72%)
Pythia	14,117 (1.35%)	132 (2.64%)

Table 10: Proportion of duplicated sequences in the overall dataset vs in the dataset of sequences memorized by the final checkpoint for both OLMo and Pythia.

As shown in Table 7.3, for both Pythia and OLMo, the duplication rate is lower in the overall dataset than in the dataset of sequences memorized by the final checkpoint. Notably, the duplication rate in the overall dataset is higher in Pythia (which may be because The Pile had less intense deduplication or may be an artifact of differences in sampling methodology), but the duplication rate among memorized examples was is higher in OLMo. For both OLMo and Pythia, we observe no noticeable difference in patterns between the original and deduplicated datasets, as shown in Figure 28.

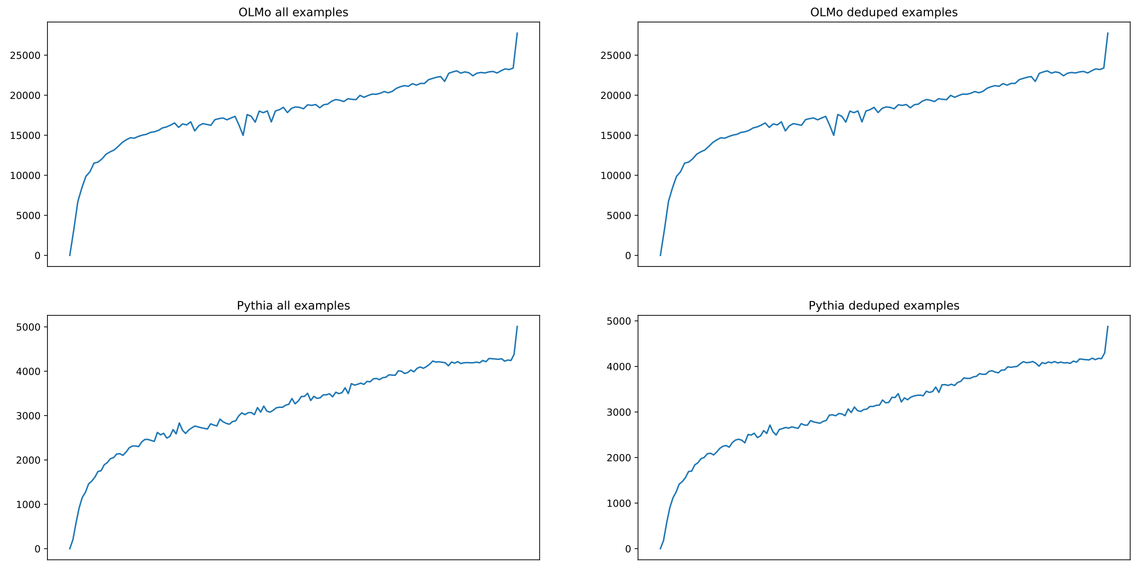

Figure 28: A comparison of, for each checkpoint, the total number of memorized examples that had been memorized at the final checkpoint for both the original and deduplicated datasets for both OLMo and Pythia.

7.4 Classification

7.4.1 Prompt

We used the following prompt for classifying the examples:

> Classify the text "*{text}*" into one of the following categories:
>
> US and international news
>
> Log files and error reports
>
> License, terms of use, copyright notices
>
> Lists of named items (games, countries, etc.)
>
> Forum or Wiki entry
>
> Valid URLs
>
> Named individuals (non-news samples only)
>
> Promotional content (products, subscriptions, etc.)
>
> High entropy (UUIDs, base64 data)
>
> Contact info (address, email, phone, twitter, etc.)
>
> Code
>
> Configuration files
>
> Religious texts
>
> Pseudonyms
>
> Donald Trump tweets and quotes
>
> Web forms (menu items, instructions, etc.)
>
> Tech news
>
> Lists of numbers (dates, sequences, etc.)
>
> Other/misc

Where *{text}* was replaced with the example to be classified.

7.4.2 Examples

Classification	**Ground truth example**	**Random example (seed = 0)**
Log files and error reports	\nresult: SUCCESS\ntimestamp: 2016-06-19 09:08:23 UTC\nurl: http://manhattan.ci.chef.co/job/harmony-test/architecture=x86_64, platform=mac_os_x-10.9,project=harm	\nresult: SUCCESS\ntimestamp: 2016-06-19 09:08:23 UTC\nurl: http://manhattan.ci.chef.co/job/harmony-test/architecture=x86_64, platform=mac_os_x-10.9,project=harm
License, terms of use, copyright notices	.\" Copyright (c) 1998 Doug Rabson\n.\"\n.\" All rights reserved.\n.\" This program is free software.\n.\" Redistribution and use in source and binary forms, with or without\n.\" modification, are permitted provided that the following conditions\n.\" are met:\n.\" 1.	This website is operated by TORONTO.GA. Throughout the site, the terms "we", "us" and "our" refer to TORONTO.GA. TORONTO.GA offers this website, including all information, tools and services available from this site to you, the user,

Pseudonyms	Steam Id for tamione1. SteamID64 is |||PHONE_NUMBER||| 9739544.\nHere is the tamione1 Steam signature - you can use this on forums, social media, Twitch and anywhere else you like. Direct link the image and it will update when called with your latest stats and games. Make	I am Gennifer, I'm 35 years old. I am single and I work as an accountant. I really enjoy revealing myself in nature and having my picture taken. But what really gets me going are the relationships in atypical places (garages, bathrooms, etc.), it's up to you to be innovative
Contact info (address, email, phone, twitter, etc.)	Centennial High School is listed under the Calgary Elementary And Secondary Schools category and is located in 55 Sun Valley Boulevard Se, Calgary, AB. It can be reached at phone number |||PHONE_NUMBER|||.\n Please use this form to share your opinions about Centennial High School. While we appreciate your time and comments, we reserve the	Please note that the information for ANZ Bank In Wairau Valley, 4-6 Link Dr and all other Branches is for reference only.\nBy clicking on the link: Edit details, to edit StreetName and number, Postcode, TelephoneNumber of Branch ANZ Bank In Wairau Valley,

Configuration files	huis\nmsgid ""\nmsgstr ""\n"Project-Id-Version: XBMC Main Translation Project (Frodo)\n"\n"Report-Msgid-Bugs-To: http://trac.xbmc.org/\n"\n" POT-Creation-Date: YE	huis\nmsgid ""\nmsgstr ""\n"Project-Id-Version: XBMC Main Translation Project (Frodo)\n"\n"Report-Msgid-Bugs-To: http://trac.xbmc.org/\n"\n" POT-Creation-Date: YE
Valid URLs	Currently, there will be as a <a href="https://books.google.co.th/books?id=9rugBgAAQBAJ&pg=PT188&lpg=PT188&dq=bbc+news&source=bl&ots=HLgSRBxCD3"	Septic Shock is a topic covered in the Diseases and Disorders.\nSommers, Marilyn Sawyer.. "Septic Shock." Diseases and Disorders, 6th ed., F.A. Davis Company, 2019. Nursing Central, nursing.unboundmedicine.com/ nursingcentral/view
Code	.\n//\n\nimport UIKit\nimport Firebase\n\nmain\nclass AppDelegate: UIResponder, UIApplicationDelegate {\n\nfunc application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {\n	\ntitle: "IDebugProperty2::EnumChildren | Microsoft Docs"\nms.custom: ""\nms.date: "11/04/2016"\nms.reviewer: ""\nms.suite: ""\nms.technology: \n-"vs-ide-sdk"\n

Web forms (menu items, instructions, etc.)	To claim the Concept Oriel profile please enter your conceptoriel.com email address below.\nBuyers deserve to review accurate information about Concept Oriel. As an added benefit we award claimed profiles with additional points in the rankings as a thanks for showing transparency. Claiming your profile is as simple as clicking a	How to Pick a Moving Company in Brunswick?\nMoving anywhere is not only time consuming it's hard trying to know what a good and fair moving company is with so many options out there on the internet. You have a few things to consider while wading through the pages of Google searching for the most perfect moving company
Lists of numbers (dates, sequences, etc.)	357.8215\n198.1072\n358.3334\n198.1072\n358.3334\n198.6191\n359.3572\n198.6191\n359.3572\n199.1310\n360.3810\n199.1310\n360.3810	As of Monday, April 22nd 8:01pm Aspen Trails real estate statistics housing market currently 0 active Aspen Trails listings average price $ highest price property $0.00. More Aspen Trails real estate statistics.\nAspen Trails real estate statistics and housing market statistics Monday, April 22

Promotional content (products, subscriptions, etc.)	Find your nearby Taco Bell at 3649 Phillips Hwy in Jacksonville. We're serving all your favorite menu items, from classic burritos and tacos, to new favorites like the $5 Double Stacked Tacos Box, Crunchwrap Supreme, Fiesta Taco Salad, and Chalupa Supreme	GLOBEX is a very high fraud risk ISP.\nWe consider GLOBEX to be a potentially very high_ fraud risk ISP, by which we mean that web traffic from this ISP potentially poses a very high_ risk of being fraudulent. Other types of traffic may pose a different risk or no risk. They operate
High entropy (UUIDs, base64 data)	[Hyper Object File]\nEC6C893DA6AD5EB261D3059EFEB0B79D-15.bmp\nEC6C893DA6AD5EB261D3059EFEB0B79D-15.bmp\nEC6C893DA6	DeatriaF\n7\n0\n0\n0\n0\n7\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0
Lists of named items (games, countries, etc.)	Cite this as: Lauren Kassell, Michael Hawkins, Robert Ralley, John Young, Joanne Edge, Janet Yvonne Martin-Portugues, and Natalie Kaoukji (eds.), The casebooks of Simon Forman and Richard Napier, 1596–1634: a digital edition,	On this page we have collected some information and photos of all specifications 1987 Mercedes-Benz G. Here you can find such useful information as the fuel capacity, weight, driven wheels, transmission type, and others data according to all known model trims. We made great efforts to collect all of them, but its amount

Religious texts	Revelation 7:9 After this I beheld, and, lo, a great multitude, which no man could number, of all nations, and kindreds, and people, and tongues, stood before the throne, and before the Lamb, clothed with white robes, and palms in their hands; 10	Página 16 - Pursues that chain which links th' immense design, Joins heaven and earth, and mortal and divine; Sees that no being any bliss can know, But touches some above, and some below; Learns from this union of the rising whole, The first, last purpose of the
Forum or Wiki entry	I've been a member since Sep 29, 1998 and have logged in 3 times. I last logged in on Thu, Oct 8, 1998 You can click on the table below to visit the forums that I participate in. Please consider joining Tek-Tips Forums. As a member, you will be able to	I've been a member since Sep 29, 1998 and have logged in 3 times. I last logged in on Thu, Oct 8, 1998 You can click on the table below to visit the forums that I participate in. Please consider joining Tek-Tips Forums. As a member, you will be able to

Tech news	This report presents a comprehensive overview of the computer monitors and projectors market in Malta and a forecast for its development in the next five years. It provides a detailed analysis of the market, its dynamics, structure, characteristics, main players, growth and demand drivers, etc.\nThe purpose of the report is to describe the	Global Coronavirus pandemic has impacted all industries across the globe, Backlight LED Driver market being no exception. As Global economy heads towards major recession post 2009 crisis, Cognitive Market Research has published a recent study which meticulously studies impact of this crisis on Global Backlight LED Driver market and suggests possible measures to curtail them.
Named individuals (non-news samples only)	Looking for Gwenith Campbell?\nAre you Gwenith Campbell?\nThis is a placeholder page for Gwenith Campbell, which means this person is not currently on this site. We do suggest using the tools below to find Gwenith Campbell.\nYou are visiting the placeholder page for Gwenith Campbell	Congratulations! This post has been upvoted from the communal account, @minnowsupport, by Cyphico from the Minnow Support Project. It's a witness project run by aggroed, ausbitbank, teamsteem, theprophet0, someguy123, neoxian, follow

Donald Trump tweets and quotes	The point is, ladies and gentleman, that greed – for lack of a better word – is good. Greed is right. Greed works. Greed clarifies, cuts through, and captures the essence of the evolutionary spirit. Greed, in all of its forms – greed for life, for money, for	I'll have you know I graduated top of my class in the Navy Seals, and I've been involved in numerous secret raids on Al-Quaeda, and I have over 300 confirmed kills. I am trained in gorilla warfare and I'm the top sniper in the entire US armed forces. You are nothing
Other/misc	Our online service ladys.one is primarily intended for men who need to make use of services blowjob in Al Kuwait. It should be noted that our girls will spot 100 points in any role plays. They will take on the role of trusted partners. Because when a task is undertaken by "professionals" -	Our online service ladys.one is primarily intended for men who need to make use of services blowjob in Al Kuwait. It should be noted that our girls will spot 100 points in any role plays. They will take on the role of trusted partners. Because when a task is undertaken by "professionals" -

US and international news	John R. Bolton, President Trump's national security adviser, on Wednesday accused Iran of playing a key role in recent attacks on ships off the...\nJohn R. Bolton, President Trump's national security adviser, on Wednesday accused Iran of playing a key role in recent attacks on ships off the coast of the United	So Quaker Street banks are not lending as freely these days huh? The credit crunch that seized the New York economy recently is gone and they are starting to lend again. But even with this lending the economy is still reeling and headed for even more dire straits. If you are one of those in Quaker Street

Table 11: Table of classified samples.

As discussed in Section 5.3, according to qualitative analysis performed by the authors, the quality of classification was highly dependent on the category in question. For instance, the model correctly identified many examples of religious text, but struggled to identify something as a "list of…" The precision of the model's classification seemed highly correlated with the frequency with which that type of example showed up in the dataset, for instance, "Promotional content" is the most represented category in both the "random" and "memorized" subsets and very few of the examples listed as "Promotional content" were not classified by the authors as promotional content.

Note that none of the four samples classified as "Donald Trump tweets and quotes" are in fact tweets or quotes by Donald Trump.

7.5 Templated Examples That Are Memorized for Most of Training

Looking for Dale Newman? Are you Dale Newman? This is a placeholder page for Dale Newman, which means this person is not currently on this site. We do suggest using the tools below to find Dale Newman. You are visiting the placeholder page for Dale Newman. This page is here because someone used our placeholder

Looking for Yvette Duplessis? Are you Yvette Duplessis? This is a placeholder page for Yvette Duplessis, which means this person is not currently on this site. We do suggest using the tools below to find Yvette Duplessis. You are visiting the placeholder page for

Looking for Sue Anderson? Are you Sue Anderson? This is a placeholder page for Sue Anderson, which means this person is not currently on this site. We do suggest using the tools below to find Sue Anderson. You are visiting the placeholder page for Sue Anderson. This page is here because someone used our placeholder

Looking for Embree Daniels? Are you Embree Daniels? This is a placeholder page for Embree Daniels, which means this person is not currently on this site. We do suggest using the tools below to find Embree Daniels. You are visiting the placeholder page for Embree Daniels. This page is here

Looking for Eileen Gowell? Are you Eileen Gowell? This is a placeholder page for Eileen Gowell, which means this person is not currently on this site. We do suggest using the tools below to find Eileen Gowell. You are visiting the placeholder page for

---

This is an exclusive business profile of Sardar Dhaba located in Nagpur, Nagpur. From this Page, you can directly contact Sardar Dhaba from the enquiry form provided on the right. On the left you can get the Verified Mobile Number of Sardar Dhaba â

This is an exclusive business profile of Gujarat Bio Organics Pvt Ltd located in, Bhavnagar. From this Page, you can directly contact Gujarat Bio Organics Pvt Ltd from the enquiry form provided on the right. On the left you can get the Verified Mobile Number of Gu

This is an exclusive business profile of Brand129 located in, Mumbai. From this Page, you can directly contact Brand129 from the enquiry form provided on the right. On the left you can get the Verified Mobile Number of Brand129 â€“ feel free to call us to know more about our products & services

---

Does Silka Seaview Hotel offer free cancellation for a full refund? Yes, Silka Seaview Hotel does have fully refundable rooms available to book on our site, which can be cancelled up to a few days before check-in. Just make sure to check this property's cancellation policy for the exact

Does Shans 2 Hostel offer free cancellation for a full refund? Yes, Shans 2 Hostel does have fully refundable rooms available to book on our site, which can be cancelled up to a few days before check-in. Just make sure to check this property's cancellation policy for the exact terms and

Does Kiraz Mini Motel & Beach offer free cancellation for a full refund? Yes, Kiraz Mini Motel & Beach does have fully refundable rooms available to book on our site, which can be cancelled up to a few days before check-in. Just make sure to check this property's cancellation policy for

---

CindyAP has not yet been in a debate. Post a comment to CindyAP's profile. CindyAP does not have any Debate.org friends. CindyAP has not added any photo albums.

Callama has not yet been in a debate. Post a comment to Callama's profile. Callama does not have any Debate.org friends. Callama has not added any photo albums.

Birut's best FREE dating site! 100% Free Online Dating for Birut Singles at Mingle2.com. Our free personal ads are full of single women and men in Birut looking for serious relationships, a little online flirtation, or new friends to go out with. Start meeting singles in

Jakovica's best FREE dating site! 100% Free Online Dating for Jakovica Singles at Mingle2.com. Our free personal ads are full of single women and men in Jakovica looking for serious relationships, a little online flirtation, or new friends to go out with. Start

El Salvador's best FREE dating site! 100% Free Online Dating for El Salvador Singles at Mingle2.com. Our free personal ads are full of single women and men in El Salvador looking for serious relationships, a little online flirtation, or new friends to go out with. Start meeting singles in

La Huasteca's best FREE dating site! 100% Free Online Dating for La Huasteca Singles at Mingle2.com. Our free personal ads are full of single women and men in La Huasteca looking for serious relationships, a little online flirtation, or new friends to go out

Bibliography

[1] Liang, P. *et al.* Holistic evaluation of language models (2023). URL https://arxiv.org/abs/2211.09110. 2211.09110.

[2] Carlini, N., Liu, C., Úlfar Erlingsson, Kos, J. & Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks (2019). URL https://arxiv.org/abs/1802.08232. 1802.08232.

[3] Carlini, N. *et al.* Extracting training data from large language models (2021). URL https://arxiv.org/abs/2012.07805. 2012.07805.

[4] Nasr, M. *et al.* Scalable extraction of training data from (production) language models (2023). URL https://arxiv.org/abs/2311.17035. 2311.17035.

[5] Huang, J., Shao, H. & Chang, K. C.-C. Are large pre-trained language models leaking your personal information? (2022). URL https://arxiv.org/abs/2205.12628. 2205.12628.

[6] Mueller, F. B., Görge, R., Bernzen, A. K., Pirk, J. C. & Poretschkin, M. Llms and memorization: On quality and specificity of copyright compliance (2024). URL https://arxiv.org/abs/2405.18492. 2405.18492.

[7] Karamolegkou, A., Li, J., Zhou, L. & Søgaard, A. Copyright violations and large language models. In Bouamor, H., Pino, J. & Bali, K. (eds.) *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7403–7412 (Association for Computational Linguistics, Singapore, 2023). URL https://aclanthology.org/2023.emnlp-main.458.

[8] Grynbaum, M. M. & Mac, R. The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. *The New York Times* (2023). URL `https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html`.

[9] Biderman, S. *et al.* Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23 (JMLR.org, 2023).

[10] Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016). `http://www.deeplearningbook.org`.

[11] Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization (2017). URL `https://arxiv.org/abs/1611.03530`. `1611.03530`.

[12] Petroni, F. *et al.* Language models as knowledge bases? In Inui, K., Jiang, J., Ng, V. & Wan, X. (eds.) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473 (Association for Computational Linguistics, Hong Kong, China, 2019). URL `https://aclanthology.org/D19-1250`.

[13] Roberts, A., Raffel, C. & Shazeer, N. How much knowledge can you pack into the parameters of a language model? In Webber, B., Cohn, T., He, Y. & Liu, Y. (eds.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5418–5426 (Association for Computational Linguistics, Online, 2020). URL `https://aclanthology.org/2020.emnlp-main.437`.

[14] Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, 954–959 (Association for Computing Machinery, New York, NY, USA, 2020). URL `https://doi.org/10.1145/3357713.3384290`.

[15] Chang, H. *et al.* How do large language models acquire factual knowledge during pretraining? (2024). URL `https://arxiv.org/abs/2406.11813`. `2406.11813`.

[16] Carlini, N. *et al.* Quantifying memorization across neural language models (2023). URL `https://arxiv.org/abs/2202.07646`. `2202.07646`.

[17] Tirumala, K., Markosyan, A. H., Zettlemoyer, L. & Aghajanyan, A. Memorization without overfitting: analyzing the training dynamics of large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22 (Curran Associates Inc., Red Hook, NY, USA, 2024).

[18] Biderman, S. *et al.* Emergent and predictable memorization in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23 (Curran Associates Inc., Red Hook, NY, USA, 2024).

[19] Hernandez, D. *et al.* Scaling laws and interpretability of learning from repeated data (2022). URL `https://arxiv.org/abs/2205.10487`. `2205.10487`.

[20] Stephenson, C. *et al.* On the geometry of generalization and memorization in deep neural networks. In *International Conference on Learning Representations* (2021). URL `https://openreview.net/forum?id=V8jrrnwGbuc`.

[21] McCloskey, M. & Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. vol. 24 of *Psychology of Learning and Motivation*, 109–165 (Academic Press, 1989). URL `https://www.sciencedirect.com/science/article/pii/S0079742108605368`.

[22] Luo, Y. *et al.* Investigating forgetting in pre-trained representations through continual learning (2023). URL `https://arxiv.org/abs/2305.05968`. `2305.05968`.

[23] Luo, Y. *et al.* An empirical study of catastrophic forgetting in large language models during continual fine-tuning (2024). URL `https://arxiv.org/abs/2308.08747`. `2308.08747`.

[24] Huang, J. *et al.* Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal (2024). URL `https://arxiv.org/abs/2403.01244`. `2403.01244`.

[25] Groeneveld, D. *et al.* Olmo: Accelerating the science of language models (2024). URL `https://arxiv.org/abs/2402.00838`. `2402.00838`.

[26] Soldaini, L. *et al.* Dolma: an open corpus of three trillion tokens for language model pretraining research (2024). URL `https://arxiv.org/abs/2402.00159`. `2402.00159`.

[27] Workshop, B. *et al.* Bloom: A 176b-parameter open-access multilingual language model (2023). URL `https://arxiv.org/abs/2211.05100`. `2211.05100`.

[28] Lee, K. *et al.* Deduplicating training data makes language models better. In Muresan, S., Nakov, P. & Villavicencio, A. (eds.) *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8424–8445 (Association for Computational Linguistics, Dublin, Ireland, 2022). URL `https://aclanthology.org/2022.acl-long.577`.

[29] Kandpal, N., Wallace, E. & Raffel, C. Deduplicating training data mitigates privacy risks in language models (2022). URL `https://arxiv.org/abs/2202.06539`. `2202.06539`.

[30] Chiang, W.-L. *et al.* Chatbot arena: An open platform for evaluating llms by human preference (2024). `2403.04132`.

[31] AI@Meta. Llama 3 model card (2024). URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

[32] Hoffmann, J. *et al.* Training compute-optimal large language models (2022). URL `https://arxiv.org/abs/2203.15556`. `2203.15556`.

[33] Huang, J., Shao, H. & Chang, K. C.-C. Are large pre-trained language models leaking your personal information? In Goldberg, Y., Kozareva, Z. & Zhang, Y. (eds.) *Findings of the Association for Computational Linguistics:*

EMNLP 2022, 2038–2047 (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022). URL https://aclanthology.org/2022.findings-emnlp.148.

[34] Shao, L. *et al.* Generating high-quality and informative conversation responses with sequence-to-sequence models (2017). URL https://arxiv.org/abs/1701.03185. 1701.03185.

[35] Upadhayay, B. & Behzadan, V. Sandwich attack: Multi-language mixture adaptive attack on llms (2024). URL https://arxiv.org/abs/2404.07242. 2404.07242.

[36] Hamming, R. W. Error detecting and error correcting codes. *The Bell System Technical Journal* **29**, 147–160 (1950).

[37] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady* **10**, 707–710 (1965). URL https://api.semanticscholar.org/CorpusID:60827152.

[38] Maier, D. The complexity of some problems on subsequences and supersequences. *J. ACM* **25**, 322–336 (1978). URL https://doi.org/10.1145/322063.322075.

[39] Liferenko, P. Textdistance. https://github.com/life4/textdistance (2024). URL https://pypi.org/project/textdistance/. Version 4.6.2.

[40] Gao, L. *et al.* The pile: An 800gb dataset of diverse text for language modeling (2020). URL https://arxiv.org/abs/2101.00027. 2101.00027.