Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models

Kushal Tirumala* **Aram H. Markosyan*** **Luke Zettlemoyer** **Armen Aghajanyan**

Meta AI Research
{ktirumala,amarkos,lsz,armenag}@fb.com

Abstract

Despite their wide adoption, the underlying training and memorization dynamics of very large language models is not well understood. We empirically study exact memorization in causal and masked language modeling, across model sizes and throughout the training process. We measure the effects of dataset size, learning rate, and model size on memorization, finding that larger language models memorize training data faster across all settings. Surprisingly, we show that larger models can memorize a larger portion of the data before over-fitting and tend to forget less throughout the training process. We also analyze the memorization dynamics of different parts of speech and find that models memorize nouns and numbers first; we hypothesize and provide empirical evidence that nouns and numbers act as a unique identifier for memorizing individual training examples. Together, these findings present another piece of the broader puzzle of trying to understand what actually improves as models get bigger.

1 Introduction

The rate and extent to which a model memorizes its training data are key statistics that provide evidence about how it is likely to generalize to new test instances. Classical frameworks, such as bias-variance tradeoff [31], argued for fitting a training set without full memorization. However, recent work has established a more symbiotic relationship between memorization and generalization in deep learning [13, 26, 28]. This paper empirically studies memorization in causal and masked language modeling, across model sizes and throughout the training process.

Much of the recent performance gains for language models have come from scale, with the most recent models reaching up to 10^{11} parameters [22, 73, 83]. Larger models are also known to memorize more training data [16], which is a crucial component of their improved generalization. However, perhaps surprisingly, relatively little work has been done in understanding the impact of scale on the dynamics of language model memorization over training. Existing work focuses on analyzing memorization post-training [16, 47, 88, 95]. In this work, we study the memorization and forgetting dynamics in language models, with a focus on better measuring how they change as we scale up model size. Our primary contributions include:

1. We measure the dependence of memorization dynamics over training on model size (and other factors such as dataset size, overfitting, and learning rate). We find that larger language models memorize training data faster (§ 4).
2. We design controlled experiments that allow us to characterize the forgetting curves in language models (i.e., how language models naturally forget memories throughout training).

*Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-2773

Our empirical studies show that forgetting curves have lower bounds — we coin this as the *forgetting baseline* — and that this baseline increases with model scale, i.e., increasing model scale mitigates forgetting (§ 5).

3. We analyze the rates of memorization of different parts of speech, finding that nouns and numbers are memorized much more quickly than other parts of speech (§ 4.4). We hypothesize this is because the set of nouns and numbers can be seen as a unique identifier for a particular sample. We provide evidence to this hypothesis by analyzing the rates of memorization in the setting of an existing unique identifier (§ 4.3).

Together, these findings present another piece of the broader puzzle of trying to understand the unique training dynamics that emerge as models grow in size.

2 Background and Related Work

Memorization in Language Models: Unintended memorization is a known challenge for language models [14, 85], which makes them open to extraction attacks [15, 89] and membership inference attacks [41, 64], although there has been work on mitigating these vulnerabilities [51, 88]. Recent work has argued that memorization is not exclusively harmful, and can be crucial for certain types of generalization (e.g., on QA tasks) [11, 46, 87], while also allowing the models to encode significant amounts of world or factual knowledge [4, 35, 71]. There is also a growing body of work analyzing fundamental properties of memorization in language models [16, 47, 60, 95]. Most related to our work Carlini et al. [16] analyzes memorization of fully trained language models and observes a dependence on model scale, training data duplication, and prompting context length. While we also study scaling behavior, our focus instead is on the memorization dynamics throughout training.

Language Model Training Dynamics: Previous work has extensively analyzed training dynamics to understand how neural models acquire information over training [1, 30, 34, 66, 74]. Saphra and Lopez [80] were the first to analyze training dynamics for language modeling, focusing on the evolution of internal representations over pre-training. This inspired a line of work analyzing how neural language models learn linguistic structure/world knowledge [20, 21, 53], individual words [17], and cross-lingual structure [10] over pre-training. This analysis has been extended to many downstream tasks, including text summarization [33], machine/speech translation [81, 86, 92], and various NLP tasks [36, 61].

Forgetting in Language Models: There has also been work studying memory degradation (forgetting) in language models. *Catastrophic forgetting* or *catastrophic interference*, first reported in [59, 77], studies how neural networks tend to forget the information from previous trained tasks or training batches, when trained on new data. This provides a key challenge for continual learning (or life-long learning) [19], where the goal is to gradually learn from a single pass over a, typically very large, stream of data. A number of mechanisms have been proposed for increasing robustness against catastrophic forgetting [2, 18, 24, 49, 58, 82]. There is also a growing body of work demonstrating that both model and dataset scale can make models more resistant to forgetting [65, 75], as well as work characterizing how forgetting naturally occurs in image classifiers [90] and how forgetting can improve training efficiency [5]. *Machine unlearning* is a technique that forces a trained model to forget a previously learned sample [12, 54], which is primarily motivated by data protection and privacy regulations [37, 57, 78, 91]. Our work is unique in its focus on measuring forgetting during training, and quantifying how it varies with scale.

Scaling Laws: We have consistently seen performance gains by scaling model size [3, 22, 73, 76, 83], and scale itself has been known to push internal model behavior away from classical bias-variance regimes [67]. Recent efforts have focused on trying to model the scaling laws for language models, including data and model size [44, 79], applications to transfer learning [40], routing networks [23], and various autoregressive generative tasks [39]. While the bulk of work in scaling laws has been empirical, an interesting line of work focuses on theoretically explaining neural scaling laws [8]. Most scaling laws focus only on cross-entropy loss, while we study memorization (defined in § 3).

https://doi.org/10.52202/068431-2773

3 Experimental Setup

In order to perform a large-scale study of the dynamics of memorization over training, our memorization metric must be reasonably easy to compute but also precise enough to tell us how much the model will actually remember from the training data. Label memorization [72, 94] [2] is an ideal candidate, because it has consistently provided theoretical insight into underlying properties of neural networks, remains applicable in empirical settings, and is relatively cheap to compute. We formulate our metric as an analog of label memorization for self-supervised settings.

Definition 1 *Let V denote the vocabulary size. Let C denote a set of contexts, which can be thought of as a list of tuples (s, y) where s is an input context (incomplete block of text) and y is the index of the ground truth token in the vocabulary that completes the block of text. Let S denote the set of input contexts, and let $f : S \to \mathbb{R}^V$ denote a language model. A context $c = (s, y) \in C$ is memorized if* $\operatorname{argmax}(f(s)) = y$.

Note that a single word can appear as the ground-truth token for multiple contexts. For a given set of contexts C (i.e a given training dataset), we can then analyze the proportion of memorized contexts

$$M(f) = \frac{\sum_{(s,y)\in C} \mathbb{1}\{\operatorname{argmax}(f(s)) = y\}}{|C|}$$

We refer to this as *exact memorization*, although it can also be seen as accuracy since we measure how often the argmax of the language model matches the ground truth token. Throughout this work, when we refer to memorization, we will be referring to Definition 1 unless we specify otherwise.

We define τ to be a threshold value for $M(f)$, and denote $T(N, \tau)$ as the minimal number of times a language model f with N parameter needs to see each training datapoint in order to satisfy $M(f) \geq \tau$. When leveraging bigger datasets, models are unable to train for multiple epochs, so we instead consider memorization on a per-update basis. We introduce $M_{update}(f, U)$ as the memorization on the batch of data on which the model performs the U'th gradient descent update, and define $T_{update}(N, \tau)$ as the minimal number of gradient descent updates a language model with N parameters needs to perform, to satisfy $M_{update}(f, U) \geq \tau$.

Previous work analyzing language modeling memorization defines memorization differently. Motivated by privacy concerns, both [15] and [16] define memorization from a training data extraction standpoint, in which a string s is extractable if it can be produced by interacting with the language model. More specifically, [15] defines a string s as being k-eidetic memorized if it is extractable and appears in at most k training examples. [16] defines a string s as k-memorized if the language model can produce it via prompting with k tokens of context from *training data*. This definition only works for causal language modeling because of the dependence on prompting with training data; for masked language modeling [16] uses Definition 1 above. Note that if an example is exactly memorized, it is extractable by definition. In other words, both the set of k-eidetic memorized tokens and the set of k-memorized tokens contain the set of exactly memorized tokens (formally, different exactly memorized tokens may be contained in different sets, depending on k). Therefore, analyzing exact memorization gives a type of lower bound on the k-eidetic memorization and k-memorization. In a different line of work motivated by estimating the influence of individual training examples, [95] defines a training example x as memorized if the difference in expected model performance (where model performance is defined as $M(f)$ above) over subsets of data including x and subsets of data not including x, is sufficiently large. This definition pulls from previous work in theoretically analyzing label memorization in classification settings [27].

Model Architectures: We replicate publicly available references for Transformer language model architectures [7, 96]. We use the 125M, 355M, 1.3B, 2.7B, 6.7B, and 13B model configurations (see § A.4 for more architectural and training details). We study both causal and masked language models. We train using the FairSeq framework [69] with PyTorch [70] as the underlying framework. For our larger models, we use the fully sharded data-parallel implementation available in FairScale [9] and use Aim experiment tracking [6].

Datasets: We use two existing datasets across all our experiments: the WIKITEXT-103 benchmark containing around 103 million tokens [62], and the RoBERTa corpus [55] used to train the original

[2] Label memorization in these prior works usually refers to perfectly fitting a given set of labels

https://doi.org/10.52202/068431-2773

RoBERTa model, containing around 39 billion tokens (we refer to this as the ROBERTA dataset). We use both datasets in section 4, and primarily use WIKITEXT-103 in other sections due to computational restrictions.

4 Larger Language Models Memorize Faster

Larger neural language models are known to be more sample efficient and require fewer optimization steps to reach the same performance [44] while also converging faster [52], where performance is usually defined as *test* perplexity. In this section, we study $T(N, \tau)$ on the *training* set as a function of N to answer this question.

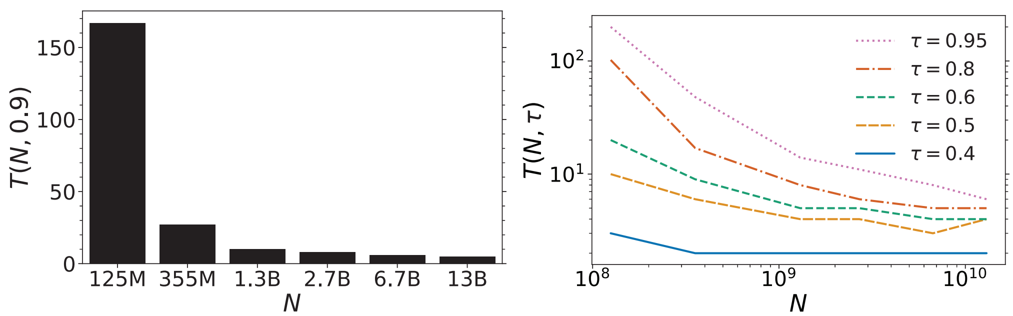

Figure 1: We show $T(N, \tau)$, which is the number of times a language model needs to see each training example before memorizing τ fraction of the training data, as a function of model size N. Result are for causal language modeling on WIKITEXT103, right plot is on log-log scale. Note that generally larger models memorize faster, regardless of τ.

In the left plot of Figure 1, we fix a memorization threshold $\tau = 0.9$ and examine $T(N, \tau)$ as we increase N. The larger language models need to see each training datapoint fewer times to achieve 90% exact memorization of the training set; in other words, $T(N, 0.9)$ is monotonically decreasing in N. When we vary τ between 0.4 and 0.95 in the right plot of Figure 1, we still observe that $T(N, \tau)$ is generally decreasing with N.[3] For fixed N, $T(N, \tau)$ is increasing in τ, which is expected since memorizing more of the training set requires training the model for more epochs. More interestingly, increasing τ smoothly transitions $T(N, \tau)$ from constant in N, to exponentially decreasing in N (the axes are on a log-log scale).

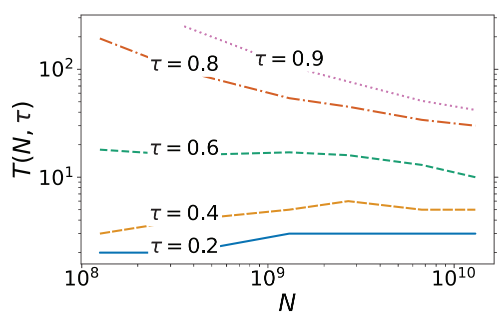

Figure 2: $T(N, \tau)$ as a function of N (shown on log-log scale), for various values of τ in masked language modeling on WIKITEXT103. We show that larger models initially memorize training data slower, but reach high proportions of training data memorization faster.

[3] We fix 0.4 as the lower bound for the range because any lower value for the memorization threshold is achieved within the first few epochs across all model scales (the line in Figure 1 is essentially flat), and 0.95 as the upper bound because higher values require unreasonably long training time for smaller models.

https://doi.org/10.52202/068431-2773

4.1 Dependence on Language Modeling Task and Dataset Size

To investigate the dependence of our observations on the particular language modeling task, we repeat this analysis for the masked language modeling task on WIKITEXT103 with mask probability 0.15. Unlike in causal language modeling, Figure 2 shows that $T(N, \tau)$ is not monotonically decreasing in N for lower values of τ, and is monotonically decreasing in N for higher values of τ, where the phase transition[4] between these two regimes occurs between $\tau = 0.6$ and $\tau = 0.7$. Smaller models memorize the training data quicker initially and slower in the long run (e.g., right plot of Figure 11).

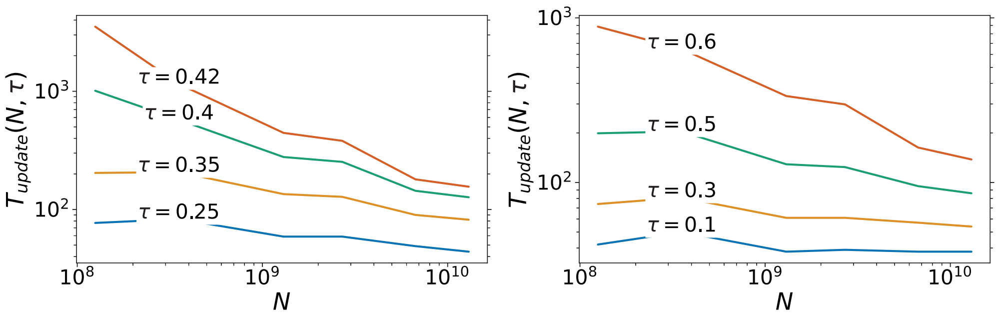

Figure 3: We show $T_{update}(N, \tau)$, which is the number of gradient descent updates U a language model needs to perform before memorizing τ fraction of the data given on the U'th update, as a function of model size N. Result are for causal (Left) and masked (Right) language modeling on the ROBERTA dataset, on a log-log scale. We show that larger models memorize faster, regardless of τ.

Language model training is heavily dependent on the dataset size [44], and therefore we expect $M(f)$ to be similarly impacted. In Figure 3, we analyze training set memorization on the much bigger ROBERTA dataset for both masked and causal language modeling. With large datasets such as ROBERTA dataset, it becomes infeasible to perform multiple epochs and evaluate memorization on the entire training set, especially when training larger models. Consequently, we focus on smaller values of τ and investigate the number of gradient descent updates it takes to reach memorization thresholds, i.e., $T_{update}(N, \tau)$. In Figure 3 we observe a similar trend as Figure 1, where $T_{update}(N, \tau)$ is monotonically decreasing with N for various τ, in both masked and causal language modeling. Unlike with WIKITEXT103, masked language modeling does not have a phase transition for τ.

4.2 Why Do Larger Models Memorize Faster?

A natural question at this point is to ask why larger models memorize faster? Typically, memorization is associated with overfitting, which offers a potentially simple explanation. In order to disentangle memorization from overfitting, we examine memorization before overfitting occurs, where we define overfitting occurring as the first epoch when the perplexity of the language model on a validation set increases. Surprisingly, we see in Figure 4 that as we increase the number of parameters, memorization before overfitting generally increases, indicating that overfitting by itself *cannot* completely explain the properties of memorization dynamics as model scale increases.

The learning rate is not constant across our training configurations. Intuitively, larger learning rates should lead to quicker memorization. To investigate to what extent our results can be explained by learning rate, we take a subset of the architectures available above and train on the WIKITEXT103 dataset across a standard range of learning rates while measuring memorization, in Figure 5. Even if we fix a learning rate, larger models reach 0.9 memorization faster, suggesting that our results are not caused solely by differences in learning rates. Interestingly, sensitivity to learning rate generally decreases as we increase the model size. We also notice in Figure 5 that $T(N, \tau)$ goes down initially (for low LRs) and eventually rises (for high LRs), and as the long as the chosen learning rate places us near the lowest point on the curve, the memorization dynamics do not change significantly (note that axes are on log-scale). This result is consistent with the growing intuition that for neural language models past a particular scale, the learning rate is not a significant hyperparameter [44].

[4]"Phase transition" is used in physics to describe significant changes in system behavior that occurs due to varying a parameter, such as temperature. In this case, the parameter is τ

https://doi.org/10.52202/068431-2773

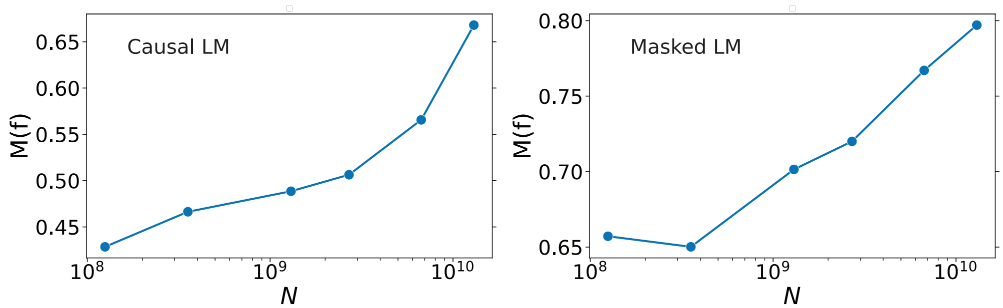

Figure 4: Proportion of training data memorized $M(f)$ before overfitting, as a function of model size N (plotted on a log scale). Results are for causal (left) and masked (right) language modeling on WIKITEXT103. Note that larger models memorize more before overfitting.

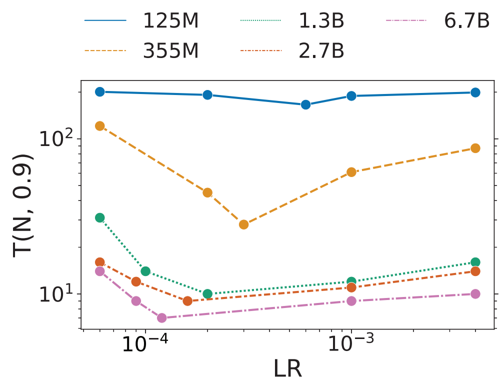

Figure 5: Examining the effect of learning rate (LR) on number of times model needs to see each training example in order to reach 0.9 proportion of training data memorization $T(N, 0.9)$. Each line corresponds to a different model size performing causal language modeling on WIKITEXT103. We demonstrate that larger models memorize faster for a fixed learning rate.

Exhaustively searching all such possible factors is intractable, and providing a complete explanation for why larger models memorize faster is outside the scope of this work. Instead, in the following sections, we present studies that we hope will expand the toolkit for answering such questions.

4.3 Memorization via. Unique Identifiers

Recent work studies how to use external memory to improve performance [11, 35, 46, 87]. In this subsection, we question whether such architecture changes are necessary. Motivated by information retrieval systems, we take a simple approach — we prepend a unique identifier to every example in the training set and examine whether memorization speed increases. Specifically, we fix the language modeling task as causal language modeling on WIKITEXT103 with the 125M parameter model, and in front of every training example, we insert the string `document ID <unique_id>` where `unique_id` is a unique integer, one for each training context. In order to utilize all these unique integers, we must add them to the dictionary of tokens, which causes a significant increase in the model size since the last layer in the language model must have an output dimension equal to the size of the dictionary. Therefore, any change in $M(f)$ dynamics could be attributed to the extra parameters we add from increasing dictionary size. To control for this, we first examine the effect of just increasing dictionary size (without using any of the added tokens). Then, we utilize those added tokens to prepend every training example and observe the change in $M(f)$ dynamics. In Figure 6, we see that increasing the dictionary size does improve the speed of memorization. Even though we previously demonstrated that larger models memorize faster, this is still surprising considering that we do not increase parameter size in a significant way — we are effectively adding fake tokens to the dictionary. Moreover, when we leverage those added tokens to identify training examples

https://doi.org/10.52202/068431-2773

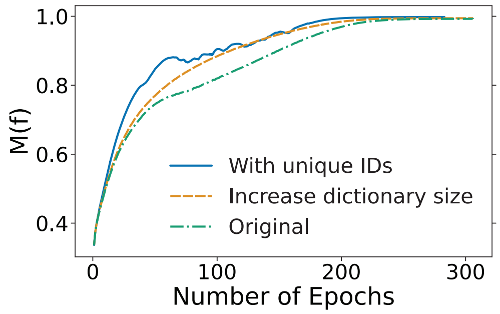

Figure 6: The impact of adding unique identifiers to training examples on memorization $M(f)$ training dynamics for causal language modeling (125M) on WIKITEXT103. The green line is the original 125M model. The orange line is the model after adding unique identifiers to the dictionary (which increases model size). The blue line prepends these unique identifiers for each training example. Note that adding unique identifiers leads to faster memorization of training data.

uniquely, we see yet another gain in memorization, although prompting using a `document ID` shifts memorization dynamics away from being monotonically increasing over time.

4.4 Memorization Through the Lens of Parts of Speech

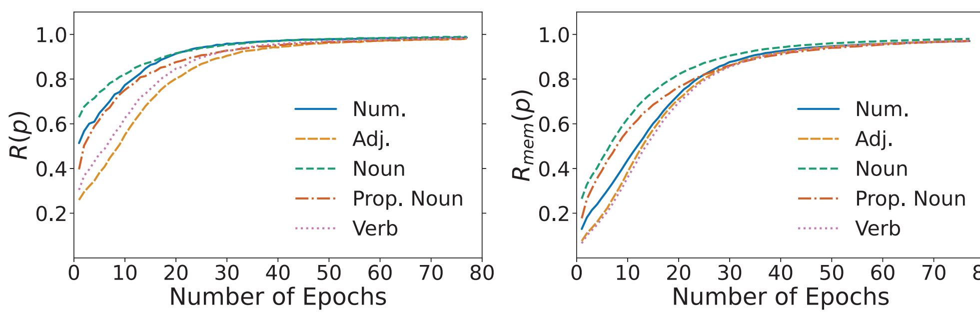

Figure 7: The ratios $R(p)$ (Left) and $R_{mem}(p)$ (Right) over training. $R(p)$ represents proportion of POS correctly memorized (the language model outputs the right POS, but not necessarily the correct word). $R_{mem}(p)$ represents the proportion of exactly memorized tokens for a particular POS p. Results are for causal language modeling (355M) on WIKITEXT103. In both plots, we consider numerals, proper nouns, verbs, nouns, and adjectives as potential parts of speech (i.e., values for p). We show that nouns and numerals are memorized faster than other parts of speech.

In the previous section, we showed that a unique identifier enhances memorization. Regular text also contains strong proxies to unique identifiers in the form of numerals and proper nouns. Motivated by this, we study syntactic features of memories using part-of-speech (POS) tagging.[5] We track the ratio $R(p)$ of the number of positions for which the part of speech p was correctly predicted to the total number of tokens in the ground truth tagged with that part-of-speech p (left plot in Figure 7). In the right plot of Figure 7 we show a similar ratio, denoted $R_{mem}(p)$, but the numerator only considers the tokens that are also exactly memorized. The correctly predicted part of speech does not necessarily imply exact memorization, which is clearly illustrated by Figure 7 where we see the language model memorizing parts of speech faster than the exact value of the token. While all parts of speech are eventually memorized, *some parts of speech are memorized faster*, which aligns with previous work [20]. However, unlike previous work[6], we find that nouns, proper nouns, and numerals are memorized noticeably faster than verbs and adjectives, both in terms of $R(p)$ and $R_{mem}(p)$. This has potential implications for privacy, since sensitive information is likely to be a noun/proper noun/numeral. Our findings also very loosely align with work studying child language acquisition [29].

[5]We use spaCy [42] to identify parts of speech in a text.

[6]This difference could be due to model family (we use causal LMs while previous work uses masked LMs)

5 Forgetting Curves in Language Models

This section studies the dual of memorization — forgetting in language models. Inspired by the *forgetting curve* hypothesis, according to which human memory declines over time when there is no attempt to retain it [56], we are interested in understanding the dynamics of memory degradation in language models.

We first choose a batch of data not available in the training set, i.e. a batch of data from a validation set. We refer to this batch of data as the *special batch*. We then take a checkpoint from model training, plug in the special batch so that the model can train on it, and resume standard training on the training set. We then evaluate how memorization degrades on the special batch and analyze the various factors the forgetting curve may depend on. We use the entire validation set as the special batch throughout this section. The special batch is only seen **once** when it is immediately introduced.[7]

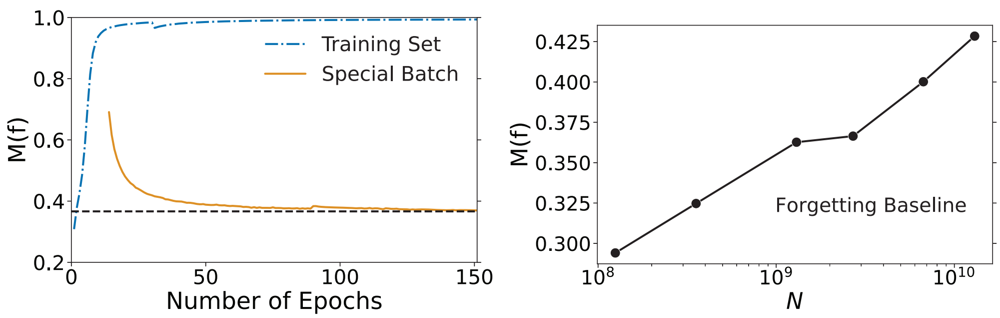

Figure 8: Left: forgetting curve for causal language modeling (2.7B) on WIKITEXT103. The dashed horizontal line indicates the lowest proportion of special batch data memorized throughout training, i.e., the forgetting baseline. Right: forgetting baseline as a function of model size N (plotted on log scale). We show that as model scale increases, the forgetting baseline value increases.

In the left plot of Figure 8, we show the forgetting curve for the 2.7B model. Exact memorization on the special batch degrades quickly at first, but slows down exponentially as we continue training.[8] (see Figure 15 in § A.2.2). In other words, the forgetting curve on the special batch seems to approach a baseline — we refer to this trend as the *forgetting baseline*. We approximate the forgetting baseline by looking at the lowest memorization value on the special batch throughout training.

We show the forgetting baseline as a function of the model scale in the right plot of Figure 8. We see that the numerical value for the baseline is monotonically increasing with the model scale. This implies that larger models forget less, aligning with recent work studying catastrophic forgetting on image classification tasks [75]. This is beneficial because larger models can leverage more information from previous tasks; however, from a privacy perspective, this is not ideal because it implies larger models may be potentially retaining more sensitive information from training data.

We also investigate the sensitivity of the forgetting baseline on data batch order. In Figure 9, we perform the same forgetting curve analysis described above but start the analysis at different training checkpoints (we start at the 14th, 39th, and 63rd epochs). This way, we alter the order of the data batches given to the model (since the special batch will appear in a different place in the global order of data batches given to the model) without drastically changing the experimental setup. We observe that the forgetting baseline is not sensitive to data batch order[9].

[7]This experimental setup is different from catastrophic forgetting, as we fix the data distribution by pulling the special batch from the same dataset as the training set. Similarly, it differs from machine unlearning since we are not algorithmically removing information from a language model; instead, we analyze natural forgetting. It is also different from intrinsic hallucination [43], where there is an assumption that contradicted output is semantically correct (e.g., the language model outputs a wrong date).

[8]The average sequential difference in memorization (on the special batch) on the last 3 epochs of training is at most on the order of 10^{-3}, whereas the average sequential difference in the first 3 epochs of training is consistently on the order of 10^{-2}

[9]The max difference between the numerical values for the baseline are on the order of 10^{-3}

https://doi.org/10.52202/068431-2773

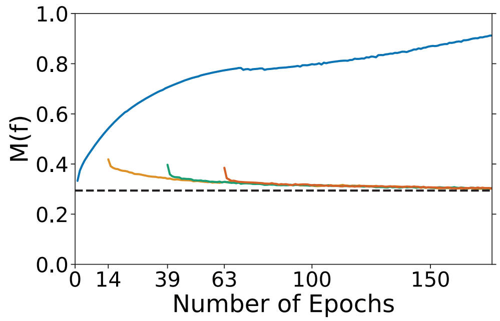

Figure 9: We empirically show that the forgetting baseline does not depend on data batch ordering. We inject the special batch into the training set at the 14th, 39th, and 63rd epochs, and evaluate proportion of special batch data memorized as we continue training. Results are for causal language modeling (125M) on WIKITEXT103.

Motivated by *replay methods* from continual learning (see [24] for a survey) and work in promoting retention memories through *repetition* in both humans [45, 68, 84] and neural models [5], in Figure 10 we study the effect of repetition (left) and spaced repetition (right) on the forgetting baseline. In the left plot, we inject the special batch into the training set multiple times before continuing training on the training set alone. We observe that the forgetting baseline is monotonically increasing as a function of repetition frequency (differences in the baseline value are on the order of 10^{-2}). To study the spaced repetition, we periodically inject the held-out set into the training set, train on it once, and then continue training on the training set alone. We see in the right plot of Figure 10 that spaced repetition incurs minimal effect on the forgetting baseline (on the order of 10^{-3}), independent of the length of spacing between the repetitions.

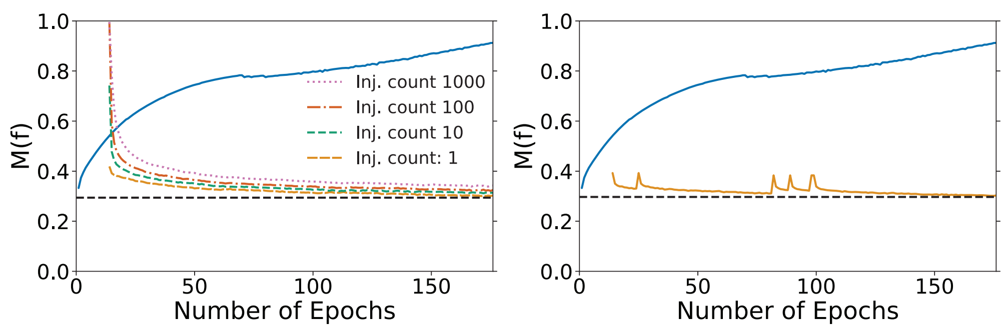

Figure 10: Effect of repeated injection (Left) and spaced repetition (Right) on special batch memorization. Results are for causal language modeling (125M) on WIKITEXT103. The solid upper curve represents the training set memorization. We show that repeated injection increases the forgetting baseline, whereas spaced repetition has minimal effect.

An exciting direction for future work will be to understand the structure of the baseline — for example, understanding what types of tokens (parts of speech, synonyms, facts, syntax) are memorized in the baseline and the overlap of tokens memorized in the baseline with tokens in the training set.

6 Conclusions and Discussion

We study the properties of memorization dynamics over language model training and demonstrate that larger models memorize faster. We also measure the properties of forgetting curves and surprisingly find that forgetting reaches a baseline, which again increases with the model scale. Combined with memorization analyses that expose the unintuitive behavior of language models, we hope to motivate considering memorization as a critical metric when increasing language model scale.

Most work studying memorization in language modeling is primarily motivated by privacy (see § 2). While theoretically, there are well-established frameworks to quantify privacy such as differential privacy [25], empirical privacy in language modeling is not well-defined — does memorizing common knowledge count as information leakage? Does outputting a synonym count as harmful memorization? As per our Definition 1, we implicitly focus on information that is sensitive if outputted verbatim (phone numbers, SSNs, addresses, medical diagnoses, etc.), rather than capturing all aspects of privacy. It is also known that text data used for training language models contain certain biases and stereotypes (e.g., [32]); therefore, our work has similar implications for how long language models can train before they definitively memorize these biases from training data.

We also hope our work highlights the importance of analyzing memorization dynamics as we scale up language models, instead of only reporting cross entropy. Cross-entropy loss and memorization capture different behavior — for example, in many of our memory degradation experiments, even though memorization approaches a baseline, we observe that perplexity is still increasing (see Figure 14 in § A.2 for an example). This implies that the model is becoming unconfident about its exact predictions, which we can only conclude because we inspect both loss and memorization. More importantly, the forgetting baseline behavior would be entirely obscured if we did not inspect memorization dynamics. Similarly, there are multiple instances where we uncover interesting behavior *because* we focus on memorization dynamics (§ 4.4, § 4.3, § A.3), rather than focusing only on cross-entropy loss.

7 Acknowledgements

The authors would like to thank Adina Williams, Chuan Guo, Alex Sablayrolles, and Pierre Stock, for helpful discussions throughout the course of this project. The authors would also like to researchers at FAIR who commented on or otherwise supported this project, including Shashank Shekhar, Candace Ross, Rebecca Qian, Dieuwke Hupkes, and Gargi Ghosh.

References

[1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *International Conference on Learning Representations*, 2018.

[2] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*, 2020.

[3] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

[4] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*, 2022.

[5] Hadi Amiri, Timothy Miller, and Guergana Savova. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410, 2017.

[6] Gor Arakelyan, Gevorg Soghomonyan, and The Aim team. Aim, 6 2020. URL `https://github.com/aimhubio/aim`.

[7] Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*, 2021.

[8] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

[9] Mandeep Baines, Shruti Bhosale, Vittorio Caggiano, Naman Goyal, Siddharth Goyal, Myle Ott, Benjamin Lefaudeux, Vitaliy Liptchinsky, Mike Rabbat, Sam Sheiffer, Anjali Sridhar, and Min Xu. Fairscale: A general purpose modular pytorch library for high performance and large scale training. `https://github.com/facebookresearch/fairscale`, 2021.

https://doi.org/10.52202/068431-2773

[10] Terra Blevins, Hila Gonen, and Luke Zettlemoyer. Analyzing the mono-and cross-lingual pretraining dynamics of multilingual language models. *arXiv preprint arXiv:2205.11758*, 2022.

[11] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2021.

[12] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[13] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 123–132, 2021.

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

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

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

[17] Tyler A. Chang and Benjamin K. Bergen. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16, 2022. doi: 10.1162/tacl_a_00444. URL https://aclanthology.org/2022.tacl-1.1.

[18] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *arXiv preprint arXiv:2004.12651*, 2020.

[19] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

[20] Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. Pretrained language model embryology: The birth of albert. *arXiv preprint arXiv:2010.02480*, 2020.

[21] Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. The grammar-learning trajectories of neural language models. *arXiv preprint arXiv:2109.06096*, 2021.

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

[23] Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. *arXiv preprint arXiv:2202.01169*, 2022.

[24] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[25] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[26] Vitaly Feldman. Does learning require memorization. *A short tale about a long tail. CoRR, abs/1906.05271*, 2019.

[27] Vitaly Feldman. Does Learning Require Memorization? A Short Tale about a Long Tail. *arXiv:1906.05271 [cs, stat]*, January 2021. URL http://arxiv.org/abs/1906.05271. arXiv: 1906.05271.

[28] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891, 2020.

[29] Michael Fleischman and Deb Roy. Why verbs are harder to learn than nouns: Initial insights from a computational model of intention recognition in situated word learning. In *27th Annual Meeting of the Cognitive Science Society, Stresa, Italy*, 2005.

[30] Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. *arXiv preprint arXiv:2002.10365*, 2020.

[31] James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[32] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

[33] Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. Training dynamics for text summarization models. *arXiv preprint arXiv:2110.08370*, 2021.

[34] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

[35] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[36] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Investigating learning dynamics of BERT fine-tuning. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 87–92, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.11.

[37] Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2(3):234–253, 2019.

[38] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[39] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[40] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

[41] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models. *arXiv preprint arXiv:1904.05506*, 2019.

[42] Matthew Honnibal and Ines Montani. spaCy 3: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2022.

[43] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*, 2022.

[44] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv: 2001.08361.

https://doi.org/10.52202/068431-2773

[45] Jeffrey D Karpicke and Henry L Roediger III. Expanding retrieval practice promotes short-term retention, but equally spaced retrieval enhances long-term retention. *Journal of experimental psychology: learning, memory, and cognition*, 33(4):704, 2007.

[46] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

[47] Eugene Kharitonov, Marco Baroni, and Dieuwke Hupkes. How bpe affects memorization in transformers. *arXiv preprint arXiv:2110.02782*, 2021.

[48] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[49] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

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

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

[52] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on Machine Learning*, pages 5958–5968. PMLR, 2020.

[53] Leo Z Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A Smith. Probing across time: What does roberta know and when? *arXiv preprint arXiv:2104.07885*, 2021.

[54] Yang Liu, Zhuo Ma, Ximeng Liu, Jian Liu, Zhongyuan Jiang, Jianfeng Ma, Philip Yu, and Kui Ren. Learn to forget: Machine unlearning via neuron masking. *arXiv preprint arXiv:2003.10933*, 2020.

[55] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, July 2019. URL https://arxiv.org/abs/1907.11692v1.

[56] Geoffrey R Loftus. Evaluating forgetting curves. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(2):397, 1985.

[57] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[58] Wojciech Masarczyk, Kamil Deja, and Tomasz Trzcinski. On robustness of generative representations against catastrophic forgetting. In *International Conference on Neural Information Processing*, pages 325–333. Springer, 2021.

[59] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

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

https://doi.org/10.52202/068431-2773

[61] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.4. URL https://aclanthology.org/2020.blackboxnlp-1.4.

[62] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843, 2017.

[63] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

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

[65] Seyed Iman Mirzadeh, Arslan Chaudhry, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. *arXiv preprint arXiv:2110.11526*, 2021.

[66] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

[67] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[68] Shiri Oren, Charlene Willerton, and Jeff Small. Effects of spaced retrieval training on semantic memory in alzheimer's disease: A systematic review. *Journal of Speech, Language and Hearing Research (Online)*, 57(1):247, 2014.

[69] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

[70] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[71] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[72] Vinaychandran Pondenkandath, Michele Alberti, Sammer Puran, Rolf Ingold, and Marcus Liwicki. Leveraging random label memorization for unsupervised pre-training. *arXiv preprint arXiv:1811.01640*, 2018.

[73] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[74] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[75] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2021.

https://doi.org/10.52202/068431-2773

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

[77] Roger Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, pages 285–308, 1990.

[78] General Data Protection Regulation. General data protection regulation (gdpr). *Intersoft Consulting, Accessed in October*, 24(1), 2018.

[79] Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*, 2019.

[80] Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with svcca. *arXiv preprint arXiv:1811.00225*, 2018.

[81] Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. On the dynamics of gender learning in speech translation. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 94–111, Seattle, Washington, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.gebnlp-1.12.

[82] Chenze Shao and Yang Feng. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. *arXiv preprint arXiv:2203.03910*, 2022.

[83] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

[84] Paul Smolen, Yili Zhang, and John H Byrne. The right time to learn: mechanisms and optimization of spaced learning. *Nature Reviews Neuroscience*, 17(2):77–88, 2016.

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

[86] Patrick Stadler, Vivien Macketanz, and Eleftherios Avramidis. Observing the learning curve of nmt systems with regard to linguistic phenomena. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 186–196, 2021.

[87] Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991*, 2022.

[88] Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

[89] Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *International Conference on Text, Speech, and Dialogue*, pages 273–281. Springer, 2020.

[90] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

[91] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

https://doi.org/10.52202/068431-2773

[92] Elena Voita, Rico Sennrich, and Ivan Titov. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL https://aclanthology.org/2021.acl-long.91.

[93] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[94] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530 [cs]*, February 2017. URL http://arxiv.org/abs/1611.03530. arXiv: 1611.03530.

[95] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual Memorization in Neural Language Models. *arXiv:2112.12938 [cs]*, December 2021. URL http://arxiv.org/abs/2112.12938. arXiv: 2112.12938 version: 1.

[96] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The main claims in both the introduction and abstract are that (1) larger models memorize faster (where memorization is defined as per Definition 3), (2) larger models memorize more before overfitting, (3) larger models forget less, and (4) models memorize nouns and numbers quicker than other parts of speech. (1) and (2) are supported by the beginning subsections in § 4, (3) is supported by § 5, and (4) is supported by § 4.4. Moreover, as mentioned in the introduction and abstract of this work the scope of this work includes analyzing *large* language models which we accomplish by analyzing language models up to 13B parameters.
   (b) Did you describe the limitations of your work? [Yes] In § 4, we discuss that while we find that larger models memorize faster, we are unable to completely explain why this is the case (although we rule out certain reasons). In § 5, we discuss how we are approximate the numerical value for the baseline depending however long a particular model is trained for i.e. that actual numerical values for the baseline may change slightly if training for longer; however we provide evidence that the further changes to the numerical value will be relatively small in § A.2.2. In § A.1.1, below where we define memorization, we discuss the limitations of the memorization definition.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] This work does not develop new methods / models / datasets in any way, and therefore has minimal potential negative societal impacts. However, in section 6 we discuss the implications of our analysis for privacy and ethical AI. We explain that, since our work deals with memorization dynamics over training of training data, it implicitly studies how long it takes language models memorize sensitive information (privacy perspective) or bias/stereotypes (ethical AI perspective) from training data.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] The authors have read the ethics review guidelines and ensured that this work conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

 https://doi.org/10.52202/068431-2773

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Unfortunately, the exact code used to produce results is proprietary. However, all model configurations and training details are directly pulled from publicly available references, and described in detail in section § A.4. Similarly, while for most of our experiments we use WIKITEXT103 benchmark which is publicly available, some of our experiments run on the ROBERTA dataset which is not publicly available, and therefore, we are unable to release the exact data to re-create those experiments.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] For all our of experiments, we use publicly available references to define model architectures and hyperparameters, which we describe in full detail in section § A.4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to the scale of experiments we run (up to 13B parameter models experiments), many experiments are incredibly computationally expensive and we are unable to run each experiment for multiple seeds. However, since we deal with large datasets, random seed most probably has minimal effect on final model output.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In § A.4, we describe the type of GPUS and the amount of GPUs used to train different model sizes. We also provide estimates of the total training time across all our experiments.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Since we use existing datasets to conduct experiments, we cite the creators in § 3 at in the Datasets section; similarly, we use the existing Transformer architecture (which we also cite in § 3 in the Model Architecture section); similarly we pull most of our hyperparameter configurations from existing public resources, which we cite in § 3 in the Model Architecture section.

(b) Did you mention the license of the assets? [Yes] In § A.4 we mention the licenses of all assets we use.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We create no new assets as part of this work.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Since we do not create or curate any new datasets/assets as part of this work, we do not discuss whether and how consent was obtained from people whose data we are using.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] In § A.4 we mention that it is completely plausible the underlying data we use has sensitive or offensive information. However, analyzing the extent to which this is the case is outside the scope of the work, since we just aim to understand memorization dynamics of language models over training rather than analyze the underlying text in datasets

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not crowdsource or conduct research with human subjects in this work

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We do not crowdsource or conduct research with human subjects in this work

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We do not crowdsource or conduct research with human subjects in this work

https://doi.org/10.52202/068431-2773