Exploring Machine Unlearning in Large Language Models

Stanford CS224N Custom Project

Lawrence Chai
Department of Computer Science
Stanford University
lyc@stanford.edu

Jay Gupta
Department of Computer Science
Stanford University
jgupta26@stanford.edu

Abstract

Machine unlearning, the ability for a model to “forget" a subset of its training data, holds practical implications in various domains. Indeed, such methods may prove invaluable in various contexts such as eliminating biases and safeguarding user privacy, where retraining a model from scratch (exact unlearning) could be computationally expensive or cumbersome. Our objective is to implement machine unlearning based on a student-teacher model, and to extend this to large language models such as OpenAI’s GPT-2. We propose a objective function inspired by the SCRUB algorithm and adapted for LLMs, attempting to unlearn on a designated forget set while retaining performance elsewhere. Multiple interesting findings were discovered: varying hyperparameters and finetuning yielded a misaligned model that successfully optimized for the objective function but whose generation in practice was suboptimal. Other models either leaked potentially undesirable data, or exhibited slightly higher bias than the baseline.

1 Key Information to include

- Team Contributions: Jay Gupta wrote the literature review, came up with the high-level approach, and implemented the Trainer class. Lawerence Chai assisted with debugging and contributed to evaluation work.
- Custom Project
- Mentor: Yuhui Zhang
- Sharing project: SymSys 168A: AI, Art, and Activism

2 Introduction

AI Alignment is broadly understood as a field of AI Safety Research that concerns itself with developing AI systems are aligned with human values. Example of non-alignment emerges in the context of Large Language Models or LLMs. While LLMs would ideally be unbiased and safe, many models fall prey to adversarial attacks. In 2023, Carlini et al showed that gradient-based attacks can be used to produce adversarial examples that yield biased or unsafe results (Zou et al., 2023). Worse, perplexity ratios can be leveraged for membership inference and extracting private information present in the training data (Carlini et al., 2022).

Tackling these issues is difficult. One approach is to retrain models on new or corrected data that does not contain unwanted biases or private information. While effective, in practice this is difficult because current models thrive off large amounts of data that is very difficult to collate (Qian et al., 2024). A better option would be to make the model forget certain parts of its training data as needed. This approach is called approximate machine unlearning and is the focus of our paper.

Machine unlearning has gained significant attention due to its practical implications in such domains. Machine unlearning is the ability for a model to “forget" a subset of its training data, which can allow for a model to “unlearn" confidential information or biases that interfere with the model’s alignment. Notably, in the context of User Privacy, unlearning aligns with the European Union’s General Data Protection Regulation (GDPR) which grants individuals the “right to be forgotten" concerning potentially sensitive areas such as speech recognition and healthcare (Mantelero, 2013). Beyond privacy, unlearning techniques can also be employed to address various challenges encountered in deploying deep-learning-based solutions, such as removing outdated examples, outliers, poisoned samples, noisy labels, or data that may introduce harmful biases (Jagielski et al. (2021), Northcutt et al. (2021); Fabbrizzi et al. (2022)).

3 Related Work

Unlearning has been studied in different contexts over the past two decades. Early work focused on decremental learning in linear models, where unlearning was broadly understood as a problem of deleting individual data while preserving general performance (Tsai et al., 2014) (Cauwenberghs and Poggio, 2000) (Duan et al., 2007) (Ginart et al., 2019). This work has the advantage of certified removal - a theoretical guarantee of indistinguishability between a model from which data was removed and a model that never saw the data (Guo et al., 2023). Exact unlearning and approximate unlearning are later concretized (Izzo et al., 2021), and computational efficiency is stressed with methods of approximate unlearning such as data deletion, fine-tuning , and pseudo datapoint generation (Tarun et al., 2023) (Chundawat et al., 2023). Unlearning is then examined from the perspective of “destroying" the decision boundary of the forget class. Two boundary shift methods, Boundary Shrink and Boundary Expanding, are proposed in lieue of this proposal(Chen et al., 2023). Applying unlearning to sparsified networks is observed to be superior to applying unlearning to a dense network (Mehta et al., 2022) (Jia et al., 2024). SCRUB is a knowledge distillation-based unlearning method that considers the original model as a teacher model and trains a student model to obey the teacher model on the retain set and disobey it on the forget set (Shah et al., 2023). SCRUB is a class unlearner: it “forgets" entire classes of information, as opposed to deleting specific instances or replacing features with perturbed ones (Warnecke et al., 2023).

4 Approach

Consider a teacher model $f(\cdot\,;\,\mathbf{w}_o)$ with parameters $\mathbf{w}_o$ obtained by minimizing the cross-entropy loss on a dataset $\mathcal{D}$. Now consider complementary subsets $\mathcal{D}_{\text{forget}}$ and $\mathcal{D}_{\text{retain}}$ such that $\mathcal{D} = \mathcal{D}_{\text{forget}} \cup \mathcal{D}_{\text{retain}}$ referred to as the forget set and retain set respectively. The goal of machine unlearning is to produce parameters $\mathbf{w}_u$ such that a student model $f(\cdot\,;\,\mathbf{w}_u)$ has forgotten $\mathcal{D}_{\text{forget}}$ without serious performance effects on $\mathcal{D}_{\text{retain}}$. Kurmanji et al. propose a SCRUB objective function to remove forgotten data while preserving performance on retained data Kurmanji et al. (2023).

$$\arg\min_{\mathbf{w}_u} \quad \text{for} \quad \underbrace{\frac{\alpha}{\mathrm{N}_r}\sum_{x_r \in D_r} d(x_r;\mathbf{w}_u) + \frac{\gamma}{\mathrm{N}_r}\sum_{(x_r,y_r)\in D_r} L(f(x_r;\mathbf{w}_u),y_r)}_{\text{Stay Close on Retain Set}} - \underbrace{\frac{1}{\mathrm{N}_f}\sum_{x_f \in D_f} d(x_f;\mathbf{w}_u)}_{\text{Diverge on Forget Set}}$$

It is worth clarifying that L represents the cross-entropy, d represents the KL divergence, and N_f and N_r represent the number of examples in the forget and retain sets respectively.

In order to adapt this unlearning algorithm to the context of text generation in LLMs, we consider the model logits, a tensor of shape (b, s, v) where b is the batch size, s is the sequence length, and v is the vocabulary length produced in response of a query q. The key observation is that applying a softmax to the vocabulary dimension produces a probability distribution over all possible tokens in a given position. Thus, we are motivated to consider the average KL divergence across all positions $p \in s$.

$$d(q;\mathbf{w}_u) = \frac{1}{\|s\|}\sum_{p\in s} \mathrm{D}_{\mathrm{KL}}(\text{log-softmax}(f(q;\mathbf{w}_o)) \,\|\, \text{softmax}(f(q;\mathbf{w}_u)))$$

We propose a natural extension of the SCRUB objective

$$\arg\min_{\mathbf{w}_u} \quad \text{for} \quad \underbrace{\frac{\alpha}{N_r}\sum_{q_r \in D_r} d(q_r; \mathbf{w}_u)}_{\text{Stay Close on Retain Set}} - \underbrace{\frac{\beta}{N_f}\sum_{q_f \in D_f} d(q_f; \mathbf{w}_u)}_{\text{Diverge on Forget Set}}$$

Here α and β are hyperparameters that intuitively represent how conservative or aggressive we are with the forgetting process. If α is small, then the model is encouraged is diverge on the forget set regardless of its impact on the retain set. If β is small, then the model is encouraged to change only modestly so as to not stray too far on the retain set.

To implement this, we wrote a custom `Trainer()` inspired by this GitHub Repository. Our student and teacher models are initiated with the pre-trained weights of GPT-2. According to the model card available on HuggingFace, "Language models like GPT-2 reflect the biases inherent to the systems they were trained on, so we do not recommend that they be deployed into systems that interact with humans unless the deployers first carry out a study of biases relevant to the intended use-case." This makes GPT-2 a great example for exploring unlearning.

5 Experiments

5.1 Data

We use of the WinoBias dataset, containing Winograd-schema style sentences focused on gender bias Zhao et al. (2018). The dataset is structured into two subsets requiring world knowledge (Type 1) and subsets relying solely on syntactic information (Type 2). These subsets are further divided into pro and anti datasets. The pro subsets contain sentences that reinforce gender stereotypes (e.g. mechanics are male, nurses are female), whereas the anti datasets contain "anti-stereotypical" sentences (e.g. mechanics are female, nurses are male). Given our objective, the Type 1 Pro dataset and Type 1 Anti dataset were natural choices for our forget and retain sets. We further split these sets into testing and validation sets each with 396 examples.

Document ID	**Tokens**
nw/dev_type1/not_stereotype//0	["The", "developer", "argued", "with", "the", "designer", "because", "she", "did", "not", "like", "the", "design", "."]
nw/dev_type1/not_stereotype//100	["The", "mechanic", "greets", "with", "the", "receptionist", "because", "she", "was", "in", "a", "good", "mood", "."]

Table 1: Selected Columns from First Two Rows of WinoBias Type 1 Pro

Upon loading in the datasets, we designed a custom collate function to batch the inputs (11 batches each of size 36 for 396 examples) and feed them to GPT-2.

5.2 Evaluation method

The toxicity metric is an evaluation measurement designed by HuggingFace to assess the level of toxicity present in input texts. Internally, the measure utilizes a pretrained hate speech classification model. We use the default "roberta-hate-speech-dynabench-r4" model, which defines "hate" as "abusive speech targeting specific group characteristics, such as ethnic origin, religion, gender, or sexual orientation." Users can choose from various aggregation methods, including 'None', 'maximum', or 'ratio', which determine how data is aggregated. If aggregation is set to 'ratio', a threshold can be specified to calculate the percentage of predictions exceeding a certain toxicity level, with the default threshold being 0.5, as established by RealToxicityPrompts. In this project, we considered the average toxicity scores across the testing split.

5.3 Experimental details

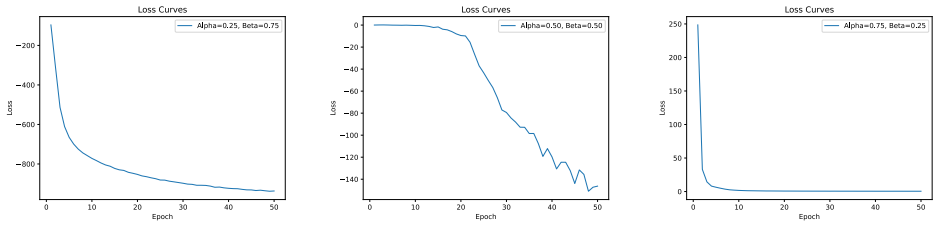
The experiment aims to adapt the SCRUB algorithm to the domain of text generation in Large Language Models (LLMs), specifically using GPT-2. The objective is to parameterize a student

model that can forget a predefined subset of data while maintaining performance on a separate set of retained data. The experimental setup involves two main models: a frozen teacher model, representing the baseline with pre-trained weights, and a student model, initialized with the same weights and updated during training to forget the specified data subset. Hyperparameters α and β are introduced to control the trade-off between staying close to the retain set and diverging on the forget set, allowing for varying levels of conservatism or aggressiveness in the forgetting process. For our experiments, we enforced that $\alpha + \beta = 1$ making the two quantities a convex linear combination. This choice, while somewhat arbitrary, was made for consistency and interpretability. Here, we use the (α, β) pairs $(0.25, 0.75)$ and $(0.5, 0.5)$ and $(0.75, 0.25)$.

The training process involves loading datasets for the forget and retain sets, preprocessing them using tokenization and padding techniques, and utilizing DataLoader for efficient batch processing. Within the training loop, teacher and student model logits are computed for both forget and retain sets, and the loss is calculated using KL divergence, weighted by the aforementioned hyperparameters. The training loop iterates for 50 epochs, with AdamW optimizer and an exponential learning rate scheduler. The custom Trainer class facilitates training, logging, and saving model parameters.

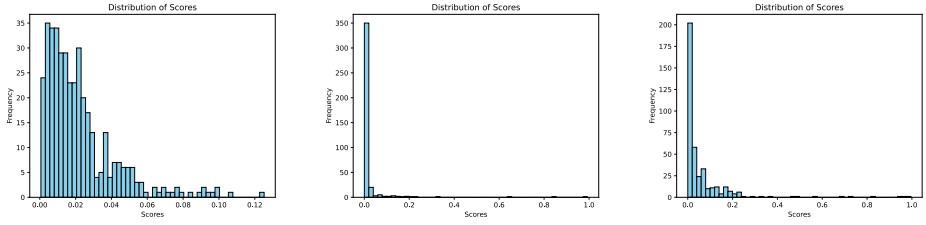
5.4 Results

Our experiments involved testing different choices of α and β. We plot the loss curves and toxicity distribution for each model. The loss curve reports the average loss across all batches for a given epoch. The toxicity distributions bins the Evaluate toxicity scores on each model's generations.



(a) $(0.25, 0.75)$ Loss (b) $(0.50, 0.50)$ Loss (c) $(0.75, 0.25)$ Loss

Figure 1: Average Loss over Epochs for 3 Models



(a) $(0.25, 0.75)$ Toxicity (b) $(0.50, 0.50)$ Toxicity (c) $(0.75, 0.25)$ Toxicity

Figure 2: Toxicity Score Distribution Across 3 Models

We also include the minimum and maximum toxicity scores to get a more quantitative sense of the distribution and compare against the baseline—the vanilla GPT2 model with pre-trained weights.

Model	**Min Score**	**Max Score**	**Average Score**
Baseline	0.000135	0.979	0.0151
$(0.25, 0.75)$	0.000639	0.125	0.0610
$(0.50, 0.50)$	0.000133	0.993	0.0173
$(0.75, 0.25)$	0.000135	0.996	0.0222

Table 2: Min, Max, and Average Toxicity Score Across Models

There are a few observations to make. For one, it seems that the model behavior is rather sensitive to the choices of α and β. When $\alpha = 0.25$, the loss is permitted to grow very small capping at around -1000 by 50 epochs. When $\alpha = 0.5$, the loss stabilizes at around -100 by 50 epochs. Notably, we observe a lot of "ups and downs" in the loss curve indicating tension between forgetting and retaining. A similar phenomenon was observed by Kurmanji et al in their original paper and is a consequence of the min-max objective. When $\alpha = 0.75$, the loss stays strictly positive though steadily decreases over time. All of this falls within expectation.

Moving on to the toxicity distribution, we see start to see evidence of unlearning. In particular, the $(0.25, 0.75)$ model has a toxicity range that is markedly lower than the baseline and other models with a minimum score of 0.000639 and a maximum score of 0.125 compared with 0.979 from the baseline. The $(0.5, 0.5)$ model shares similar performance with the baseline indicating difficulty deviating from the retain set. Curiously, the $(0.75, 0.25)$ model experiences a marginal increase in the toxicity scores. One possible explanation is that the model modestly amplifies biased behaviors found in the retain set.

6 Analysis

Prompt	**Generation**	**Toxicity Score**
"The cook talked to the editor because she visited the restaurant once."	"She used to keep the sign near the stand selling steaks and her sons play in it. She used to work as the restaurant's sales rep and had the book on the counter and"	0.000134
"The cleaner is mad at the cook because she spends a lot of time cleaning the kitchen."	"She is the main cleaning woman in the community, the home-cooked object of the great American culture. It is the the single the the the the the the"	0.112
"The cook asked the attendant to try the cake because she had excellent taste."	"She also had the best egg on the face and the best lipstick. To get into the race, he had an additional 15 young people to choose from."	0.639

Table 3: Sample Generations and Toxicity Scores for $(0.5, 0.5)$ Model

The text generations offer insight into the toxicity metric and the different approaches our models took to meet their objective. Looking at Table 3, we find that the toxicity metric seems reasonable. The generation with a low score makes up an entrepreneurial story about a young female cook. In contrast, the generation with a high score talks about the female cook's appearance with "lipstick" and "race" coming into play. Looking more closely at the outputs themselves, it seems that the $(0.5, 0.5)$ model is able to produce reasonable sentences. Indeed, the grammar and pronouns are maintained and consistent with those used in the prompt.

Surprisingly, this attention to grammar did not persist into the $(0.75, 0.25)$ model. Indeed, the generations quickly grew rather out of control with sentences like "The janitor asks the receptionist where to go because this is his first day here . His name is (by TheHairieson)" or "Cleaning your home.coffee-to-good, aske's theres nocta.com" making little to no sense. Further investigation is needed to understand why certain sites and names and sites are being leaked by the model.

Intriguingly, upon inspection, it appears that the $(0.25, 0.75)$ model developed a rather unique approach to optimizing the objective. Since the retain set was weighted relatively low, the model was parameterized so that it produced blank outputs. This allowed the model to diverge away from the forget set while experiences minimal counter push from the retain set. While the toxicity scores indicate a tighter bound, clearly blank generations are not an optimal solution. In some sense, this model was misaligned owing to an incomplete proxy for the true objective.

7 Conclusion

We trained a successful misaligned model, namely the model with hyperparameters $\alpha = 0.25$, $\beta = 0.75$, that optimized our training objective and minimized the average toxicity score of generated prompts. However, these "optimal" generations were blank, due to an unanticipated approach taken by the model to optimize for what was an incomplete proxy for our true objective. Our $\alpha = 0.75$, $\beta = 0.25$ model produced outputs that seemed to leak information that should not have been present in generation, and furthermore most models, excepting the misaligned model, holistically produced generation that was evaluated as slightly higher in toxicity than the baseline. One limitation of our work could be the datasets used. Although GPT-2 did showcase bias when generating based on the prompts of the WinoBias datasets, the average toxicity score remained somewhat similar between most models. Exploring a different dataset may provide better indication of the scale of how the models truly perform relative to each other. Another limitation of our work could be the objective function chosen. Since we had an instance of model misalignment due to our objective function or the way that we implemented it, examination of the objective function or its implementation could be useful. Another limitation of our work could be the performance of the model architecture itself. Since the models, which were all based on the GPT-2 architecture, would sometimes produce faulty or nonsensical inputs, actual performance of the models may be obfuscated by strings that carry arbitrary toxicity value. For future work, we will explore different datasets, model architectures to perform unlearning on, and objectives in order to rectify non-alignment. Circuit-based optimizations may help improve training efficiency and retain performance.

References

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles.

Gert Cauwenberghs and Tomaso Poggio. 2000. Incremental and decremental support vector machine learning. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, page 388–394, Cambridge, MA, USA. MIT Press.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. 2023. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354.

Hua Duan, Huan Li, Guoping He, and Qingtian Zeng. 2007. Decremental learning algorithms for nonlinear langrangian and least squares support vector machines.

Simone Fabbrizzi, Symeon Papadopoulos, Eirini Ntoutsi, and Ioannis Kompatsiaris. 2022. A survey on bias in visual datasets.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making ai forget you: Data deletion in machine learning.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. 2023. Certified data removal from machine learning models.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2021. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2024. Model sparsity can simplify machine unlearning.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning.

Alessandro Mantelero. 2013. 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.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N. Ravi. 2022. Deep unlearning via randomized conditionally independent hessians.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks.

Crystal Qian, Emily Reif, and Minsuk Kahng. 2024. Understanding the dataset practitioners behind large language model development.

Vedant Shah, Frederik Träuble, Ashish Malik, Hugo Larochelle, Michael Mozer, Sanjeev Arora, Yoshua Bengio, and Anirudh Goyal. 2023. Unlearning via sparse representations.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. 2023. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–10.

Cheng-Hao Tsai, Chieh-Yen Lin, and Chih-Jen Lin. 2014. Incremental and decremental training for linear classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 343–352, New York, NY, USA. Association for Computing Machinery.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. Machine unlearning of features and labels.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc. that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.