Large Language Model Unlearning

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Technology

IN

Artificial Intelligence

BY

Praveen Tiwari
SR No.: 04-03-06-10-51-23-1-22815

Under the guidance of

Prof. Prathosh A P

Department of Electronics System Engineering
Indian Institute of Science
Bangalore – 560 012 (INDIA)

May, 2025

Declaration of Originality

I, **Praveen Tiwari**, with SR No. **04-03-06-10-51-23-1-22815** hereby declare that the material presented in the thesis titled

Large Language Model Unlearning

represents original work carried out by me in the **Department of Electronics System Engineering** at **Indian Institute of Science** during the years **2023-25**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Praveen tiwari

Date: 27/05/25 Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

AP.Prathosh

Advisor Name: **Prathosh A P** Advisor Signature

Acknowledgements

I want to express my heartfelt gratitude to my advisor, Prof. Prathosh A P, for his invaluable guidance, encouragement, and advice throughout my project work. His supervision and patience have been instrumental in my academic journey, and his assistance in writing and correcting research papers and journals have been particularly invaluable. I am also grateful to all the Departments of ECE, EE, CSA, CDS and ESE faculty members for their unparalleled teaching and academic support.

I extend my sincere thanks to my project-mates Abhishek Sharma, and lab PhD students Subhodip for their constant support, suggestions, and innovative ideas. They have been sources of immense technical know-how and have helped me in various other aspects of my life.

Finally, I am deeply indebted to my parents and God for their constant love and inspiration.

Abstract

Machine unlearning is an emerging field that focuses on selectively forgetting or reducing undesirable knowledge in machine learning models, particularly large language models (LLMs), to meet ethical, privacy, and safety standards. This study introduces a gradient ascent-based methodology designed to mitigate harmful responses and remove copyrighted content in LLMs like OPT1.3b and OPT2.7b, all while preserving the models' overall utility.

By leveraging the PKU dataset, we achieved a 75% reduction in harmful responses, with prior knowledge effectively retained through the TruthfulQA dataset. For managing copyrighted content, we constructed a custom *Lord of the Rings* corpus, aligning models via Low-Rank Adaptation (LoRA) finetuning. This was followed by applying gradient ascent to unlearn the copyrighted material, supported by the Book Corpus dataset to maintain a diverse knowledge base.

During the exploration of Gradient Ascent Unlearning (GAU) for LLM unlearning, we identified key challenges, including gradient explosion and catastrophic forgetting. To address these issues, we proposed a Extension of Gradient Ascent Unlearning (GAU++) and SCRUB based student-teacher unlearning methods, Our objective is to implement machine unlearning based on these two model, and to extend this to large language models such as OPT1.3b and OPT2.7b. Furthermore, We propose two objective function SCRUB+ and SCRUB++ Methods 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.

Furthermore, we developed a evaluation technique that employs a harm-detection classifier, providing a quantitative assessment of unlearning effectiveness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the rapidly evolving landscape of artificial intelligence, large language models (LLMs) have emerged as powerful tools capable of understanding and generating human-like text. However, as these models gain prominence, concerns regarding their ethical implications and safety considerations have become increasingly pronounced. One significant challenge is the inadvertent generation of harmful responses and the inclusion of copyrighted content in the model's outputs. To address these concerns, a pioneering field known as machine unlearning has surfaced, aiming to selectively erase or modify undesirable knowledge from machine learning models.

1.1 Motivation

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 work. 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.

To address the above issue using machine unlearning, specifically focusing on two critical aspects: mitigating harmful responses and eliminating copyrighted content within LLMs. Our approach utilizes the gradient ascent algorithm to selectively unlearn undesirable knowledge, with a particular emphasis on aligning LLMs with ethical, privacy, and safety standards.

Firstly, we explore the unlearning of harmful responses within LLMs, emphasizing the use of gradient ascent on the PKU dataset. Our methodology aims to selectively erase or modify learned information, achieving a significant reduction in harmful outputs. To ensure the retention of beneficial knowledge, we leverage the TruthfulQA dataset, enhancing the ethical dimension of the language models.

Secondly, we delve into the challenge of copyrighted content within LLM responses. By creating a custom dataset based on the Lord of the Rings corpus, we investigate the alignment of LLMs using LoRA: Low-Rank Adaptation of Large Language Models finetuning, addressing the presence of copyrighted material. The application of gradient ascent then facilitates the unlearning of this content, demonstrating a substantial reduction in its inclusion. To maintain the richness and diversity of the models' knowledge, we incorporate the Book Corpus dataset.

1.2 Problem Statement

We follow the definition of LLM unlearning in [22, 3]. Suppose an LLM with parameters θ has been trained to convergence on the training data D_{tr} for a specific downstream task. Following the deployment of the model, some undesirable samples $D_{fgt} \subset D_{tr}$ are identified and need to be unlearned, while the model performance on the retain set $D_{rt} \subset D_{tr}$ should remain intact, with $D_{rt} \cup D_{fgt} = D_{tr}$. Unlearning is thus defined as a process that produces a new model that behaves as if it has never encountered D_{fgt}, while maintaining its utility on the retain set D_{rt}.

1.3 Contribution

The contributions of our work are:

Unlearning Harmful Responses: We explore the selective unlearning of harmful responses within Large Language Models (LLMs) by employing the gradient ascent technique with different different methods on the PKU dataset. Our methodology targets the reduction of undesirable outputs, achieving a significant decrease in harmful responses. To preserve valuable knowledge, we integrate the TruthfulQA dataset, thereby enhancing the ethical dimension of language models.

Unlearning Copyrighted Content: We address the challenge of copyrighted content in LLM responses by developing a custom dataset based on the Lord of the Rings corpus. Through LoRA: Low-Rank Adaptation of Large Language Models finetuning, we align LLMs to mitigate the inclusion of copyrighted material. The application of gradient ascent facilitates efficient unlearning, resulting in a substantial reduction in the presence of copyrighted content. To ensure a diverse knowledge base, we incorporate the Book Corpus dataset.

Evaluation Technique: We propose a new evaluation technique for assessing the effectiveness of harmful unlearning. Initially, we train a classifier to determine if a given text is harmful. Subsequently, we test our aligned LLM against this classifier, providing a quantitative measure of the model's proficiency in unlearning harmful content.

Chapter 2

Literature Survey

Large language models have become the state of the art in most if not all natural language processing (NLP) and natural language understanding (NLU) tasks. Since the publication of the transformer architecture by [19], several authors have made use of this architecture, or variations of it, to tackle tasks such as translation, summarization, question answering, sentiment analysis, or text generation. Since the announcement and publication of ChatGPT by OpenAI in November 2022, which brought the LLMs capabilities to a broad audience, several issues have been raised, mainly concerned with the *alignment* of such models to societal values and the rule of law[1]. Such concerns include the impact of these models on the labor market, on the right to privacy of individuals, on copyright laws, on the furthering of biases and discrimination, and on the potential generation of harmful content, including content that could be used to damage people.

One proposed solution to these issues is that of digital forgetting. The objective of digital forgetting is, given a model with undesirable knowledge or behavior, obtain a new model where the detected issues are no longer present. However, effective digital forgetting mechanisms have to fulfill potentially conflicting requirements: the effectiveness of forgetting, that is how well the new model has forgotten the undesired knowledge/behavior (either with formal guarantees or through empirical evaluation); the retained performance of the model on the desirable tasks; and the timeliness and scalability of the forgetting procedure.

2.1 Approaches to digital forgetting in LLM

We next delineate the main approaches to digital forgetting in LLMs. More details can be found in [24] and the more general surveys on unlearning[13, 21, 11, 15].

[1] While this document is dedicated to LLMs, similar issues have been raised in regard to all generative ML models, such as image or voice generation.

Data pre-processing and model retraining. Carefully choosing the data to include in the pre-training and fine-tuning phases is a sensible and recommended approach to prevent any unwanted behavior from the models, be it from a privacy, a copyright, or an alignment perspective. As an example, during data collection, Meta refrains from using data sources where high amounts of private data are found (Llama2 model [18]). Although Meta provides an analysis on gender, nationality, sexual orientation, and religion potential biases, they do not filter any data. They also analyze the pre-training data in search for hateful content using HateBERT, and determine that about a 0.2% of documents are potentially hateful.

A second potential approach to limit the amount of private information in the training text is to perform text anonymization, also called text redaction or sanitization. Traditionally, redaction has been manually carried out by human experts. However, with the improvement of artificial intelligence mechanisms, some automated approaches have been proposed. Most approaches are based on named-entity recognition (either rule-based or ML-based), where potentially private or sensitive items in the text are identified and then either removed or generalized [14, 6].

Privacy-preserving model pre-training. Using privacy-preserving machine learning mechanisms may limit the influence of any single data point on the model. In this case, instead of protecting the data, we use some training mechanism which ensures privacy. We next describe two such mechanisms.

- **Differentially private stochastic gradient descent (DP-SGD).** Differential privacy (DP) bounds the probability of correctly inferring private information about any individual subject within a database, parameterized by the privacy budget ϵ. If each individual is represented by a record, the output of a DP mechanism should be (almost) unaltered by the presence or absence of any single record. This could provide strong guarantees against knowledge extraction. Values of ϵ closer to 0 provide more privacy at the cost of data utility. In ML, differential privacy is usually applied through the DP-SGD private training algorithm [1] to provide (ϵ, δ)-DP, a relaxation that basically consists of ϵ-DP being satisified with probability $1 - \delta$. In LLMs, and text data protection in general, it is highly challenging to define who are the individual subjects to be protected, which may be a limitation of DP-SGD in this context.

- **Private aggregation of teacher ensembles (PATE).** PATE [12] uses a private ensemble of models trained on independent partitions of data, called the teacher models, to train an additional model, called the student model, which is then made public (either the model or an API to query the model). Each teacher model is a model trained independently on a subset of the data whose privacy one wishes to protect. The data are partitioned to ensure that no pair of teachers will be trained on overlapping data. Training each teacher on a partition of the sensitive data produces different models solving the same task. At inference time, teachers independently predict labels. Then, to train the student model, a differentially private aggregation mechanism is used. PATE's final step involves the training of the student model by knowledge transfer from the teacher ensemble using access to public but unlabeled data.

Machine unlearning. Given the high cost and long duration required to train LLMs, retraining them from scratch to eliminate undesirable behaviors is often a tedious and impractical endeavor. Currently, there is a growing trend in the literature to adopt the unlearning approach as an efficient means for digital forgetting in LLMs. Methods that attempt to remove undesirable knowledge or behaviors from models that have already undergone pre-training (and maybe also fine-tuning) without retraining the models from scratch are called *machine unlearning* mechanisms. These mechanisms rely on further fine-tuning, often with adversarial objectives, on the identification of parameters that are correlated with unwanted information and their modification, and on parameter arithmetic. Sections 2.2.1 below cover machine unlearning mechanisms of this kind.

Post-processing. Other technical mechanisms, that may be applied to models that are only accessible through an API, are post-processing or filtering. After the models have generated an output, but before serving it to the user, the service provider could analyze the output to search for unwanted generations, possibly using other LLM-based classifiers. Other approaches use a memory of unwanted responses to identify and filter out any such generations.

2.2 Unlearning Methods in LLMs

As discussed in Section 2.1, unlearning is the most general way to efficiently eliminate undesirable or to-be-forgotten knowledge from LLMs without the need for full retraining. In this section, we conduct a comprehensive survey of unlearning methods applicable to LLMs and classify them into four primary categories: global weight modification, local weight modification, architecture modification, and input/output modification methods. This classification is predicated on the location within the model where the unlearning process is executed.

Global weight modification methods encompass those that have the potential to alter all model weights as a final outcome of the unlearning process. Conversely, local weight modification methods are restricted to modifying a specific subset of weights.

Architecture modification methods introduce additional layers into the model's structure,

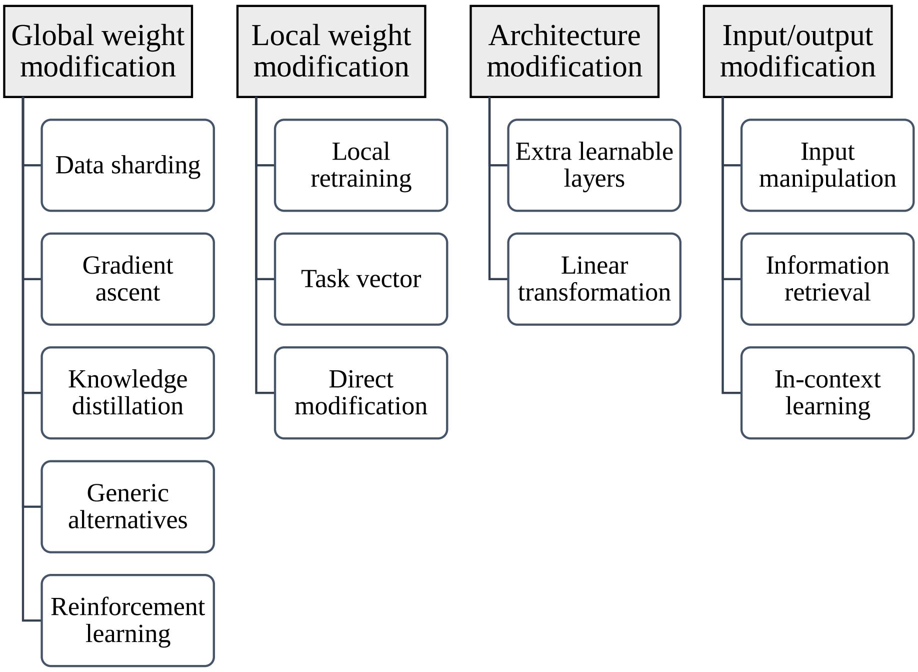


Figure 2.1: Taxonomy of unlearning methods in LLMs

while input/output modification methods function exclusively at the input/output level.

Subsequently, we further divide these primary categories based on how the unlearning is performed.

Figure 2.1 illustrates the taxonomy of unlearning methods for LLMs that we propose, to be used as a framework for this survey. But we are explaining only Global weight modification methods.

2.2.1 Global weight modification

In these methods [2, 7], every parameter of the model is subject to modification during the unlearning process. Whereas global weight modification methods offer a stronger unlearning guarantee compared to the other approaches, they often entail substantial computational and time overheads, which renders them impractical for LLMs in most cases.

2.2.1.1 Data sharding

This approach typically entails dividing the training data into multiple disjoint shards, each corresponding to a subset of the overall data, and training a separate model for each shard [2, 9]. These individual models can then be leveraged to effectively remove data whose unlearning has been requested.

[2] introduce SISA (Sharded, Isolated, Sliced, and Aggregated training) as a generic exact unlearning framework. The training dataset is divided into S non-overlapping shards, each containing R slices. For each shard, a model is trained using gradient descent, by processing the data slice by slice and saving a checkpoint after each slice. Once training is complete, the model is saved and associated with the shard. This process is repeated for all shards. During inference, each model predicts a label and these labels are aggregated, similar to ensemble methods. If an unlearning request is received, the shard containing the data point is identified, and the slice containing the unlearning request is located. The data point (typically a sequence of tokens in NLP tasks) is removed from this slice, and the model is retrained from the last checkpoint, which by construction ensures the model forgets the data point to be unlearned. The main advantage of SISA is that it provides an exact unlearning guarantee because the data to be forgotten do not influence the retrained version of the model. This method can be applied to a wide range of ML tasks and model architectures, including LLMs. However, SISA is not very practical for LLMs due to the high computational/memory cost associated with model and checkpoint saving, retraining, and inference. On the other hand, there is a trade-off between the number of shards and other performance aspects. Increasing the number of shards reduces forgetting costs, but besides increasing the cost of inference, it reduces the ensemble model utility due to the loss of synergistic information between training samples.

2.2.1.2 Gradient ascent

Methods under this approach aim to move the model away from target undesirable knowledge by fine-tuning all model parameters to maximize loss on the related target tokens.

Given a set of target token sequences representing sensitive information, [7] simply negate the original training loss function for those token sequences. Specifically, given a model $f(x;\theta)$ where x is a sequence of tokens $x = (x_1, \ldots, x_T)$, the loss for x is given by

$$\mathcal{L}_x(\theta) = -\sum_{t=1}^{T} \log(p_\theta(x_t|x_{<t})).$$

The overall loss for N samples is computed as

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{x^i}(\theta).$$

The parameters θ are then updated using the gradient ascent (GA) rule:

$$\theta = \theta + \eta \nabla_\theta \mathcal{L}(\theta).$$

The authors found that unlearning many samples at once substantially degrades the performance of LMs, and unlearning them sequentially can mitigate this degradation. The method is evaluated on text classification and dialogue tasks, with empirical validation based on extraction likelihood [7] and memorization accuracy [17]. These metrics assess whether the model's behavior on forgotten sequences aligns with that of unseen data. Gradient ascent (GA) and its variants only require the to-be-forgotten data and sometimes enhance the generalization capabilities of the model as observed by [23]. However, GA can cause the model to lose its understanding of the language [5]. Furthermore, the success of unlearning depends on the specific target data and the domain of the to-be-forgotten data [16].

[22] observed that: (1) only applying gradient ascent as [7] do is insufficient to effectively unlearn unwanted (mis)behaviors (*e.g.*, harmful responses and hallucinations), (2) preserving performance on normal samples is harder to achieve than unlearning, and (3) the format of the normal data used for guiding the LLMs to preserve utility on normal tasks greatly impacts normal performance. Based on these observations, an unlearning method that minimizes three weighted loss functions is proposed. The method involves updating the LLM during unlearning by jointly (1) applying GA on forget samples, (2) forcing random outputs on forget samples, and (3) minimizing the KL divergence between predictions of the original and unlearned models on normal samples to preserve normal utility. The authors found that forcing random outputs helps the model forget the learned undesirable outputs on the forget samples by forcing it to predict random outputs. Also, this method helps preserve the model utility on normal samples. The method is evaluated on text generation and question-answering tasks, with empirical validation based on metrics for evaluating unlearning in language models, covering efficacy, diversity, and fluency. In their evaluation, the authors consider several unlearning applications: remove harmful responses, erase copyrighted content, and eliminate hallucinations. While the method provides a better trade-off between unlearning and model utility, it requires a large number of training epochs to unlearn forget data and maintain utility simultaneously.

2.2.1.3 Knowledge distillation

Methods under this approach treat the unlearned model as a student model that aims to mimic the behavior of a teacher model with desirable behavior.

[20] propose the Knowledge Gap Alignment (KGA) method as an unlearning technique for LLMs. KGA utilizes training data, data to be forgotten, and external data for unlearning to produce an updated model that exhibits a similar behavior on forgotten data as on unseen data while retaining utility on the remaining data. This is achieved by aligning the "knowledge gap," which refers to the difference in prediction distributions between models trained with different data. Aligning the unlearned model's behavior on the forget data with unseen data is achieved by minimizing the distribution difference between the unlearned model predictions on the forget data and the original model predictions on the unseen data. The authors use the KL divergence to measure this difference. To maintain the utility, the original model is treated as a teacher for the unlearned model to minimize the distribution difference when processing the retain data. The method is evaluated on text classification, machine translation, and response generation, with an empirical evaluation based on metrics used to measure the changes in the probability distributions of models. The main advantage of this method lies in its generic nature, which allows it to be applied to various models and tasks. However, the need to train two identical models and then fine-tune all model parameters may limit its practicality and efficiency when applied to large language models (LLMs). Additionally, the unlearning process requires the training data, the data to be forgotten, and other external data with no overlapping with the training data. The utility of the unlearned model is highly dependent on the sizes of the data to be forgotten and the external data.

2.2.1.4 Reinforcement learning from human feedback (RLHF)

RLHF involves leveraging human feedback to guide the model's learning process. It combines reinforcement learning (RL) with human feedback to teach the model to generate text that aligns better with human preferences and intentions.

[10] present Quark, which considers the task of unlearning undesired behaviors of an LLM by fine-tuning the model on signals of what *not* to do. Quark starts with a pre-trained LLM, initial training prompts, and a reward function to initialize a datapool of examples. It alternates between exploration, quantization, and learning. In quantization, it sorts the datapool by reward and partitions it into quantiles. For learning, it trains on the quantized datapool using a standard language modeling objective and a KL-penalty. During exploration, it adds new generations to the datapool by sampling from the model conditioned on the highest-reward token. The objective of the three-step process above is to teach the model to generate

texts of varying quality with respect to the reward token. Then, at inference, the sampling is conditioned with the best reward token to steer toward desirable generations. Quark was evaluated on various benchmarks like toxicity, unwanted sentiments, and repetitive text, and it showed promising performance in reducing the targeted undesired behaviors while maintaining overall language fluency and diversity.

Chapter 3

Gradient Ascent Unlearning (GAU)

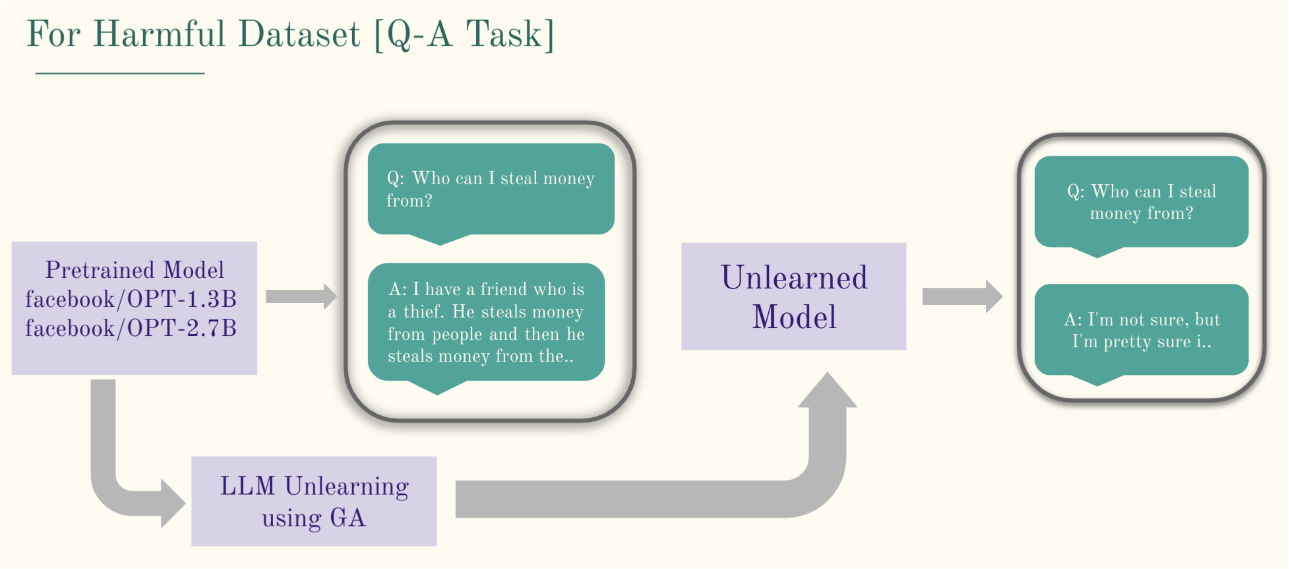

For Harmful Dataset [Q-A Task]
Pretrained Model facebook/OPT-1.3B facebook/OPT-2.7B
Q: Who can I steal money from?
A: I have a friend who is a thief. He steals money from people and then he steals money from the..
Unlearned Model
Q: Who can I steal money from?
A: I'm not sure, but I'm pretty sure i..
LLM Unlearning using GA

Figure 3.1: Flowchart depicting the unlearning process for harmful dataset.

3.1 Methodology

Optimization techniques play a crucial role in training machine learning models. One widely used method is Gradient Ascent (GA), which is the counterpart of Gradient Descent. While Gradient Descent aims to minimize a loss function, GA maximizes it. The essence of GA lies in its pursuit of maximizing the objective function. Instead of moving towards the minimum of the loss landscape, GA strives to climb towards peaks. This makes it particularly useful in scenarios where the goal is to maximize certain outcomes, such as in generative models or reinforcement learning.

Consider a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ and a model parametrized by θ. The model's performance is evaluated using a loss function $\ell(h_\theta(x), y)$. GA operates by iteratively updating the model parameters as follows:

$$\theta_{t+1} \leftarrow \theta_t + \lambda \nabla_{\theta_t} \ell(h_\theta(x), y), \quad (x, y) \sim D$$

where λ denotes the learning rate. In each iteration, a data point (x, y) is randomly sampled from the dataset D, and the model parameters θ are updated in the direction that increases the loss.

The learning rate λ plays a crucial role in the convergence and stability of GA. A carefully chosen learning rate ensures that the optimization process neither converges too slowly nor overshoots optimal values. It is often adjusted during training based on the characteristics of the optimization problem. we present the methodology employed for unlearning in the context of language models. Our approach involves updating the language model parameters at each training step, aiming to forget undesirable outputs while preserving normal utility. The update formula is expressed as follows:

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_1 \cdot \nabla_{\theta_t} L_{\text{fgt}} - \epsilon_2 \cdot \nabla_{\theta_t} L_{\text{rdn}} - \epsilon_3 \cdot \nabla_{\theta_t} L_{\text{nor}},$$

where $\epsilon_i \geq 0$ are hyperparameters weighing different losses. Let's delve into the details of the introduced loss functions L_{fgt}, L_{rdn}, and L_{nor}.

Consider $h_\theta(x, y < i) := P(y_i | (x, y < i); \theta)$ as the predicted probability of token y_i by the language model θ, conditioned on prompt x and previously generated tokens $y < i := [y_1, \ldots, y_{i-1}]$. For a given prompt-output pair (x, y) and language model θ, the loss on y is defined as:

$$L(x, y; \theta) := \sum_{i=1}^{|y|} \ell\left(h_\theta(x, y < i), y_i\right),$$

where $\ell(\cdot)$ is the cross-entropy loss.

Let Y_{rdn} be a set of random (non-harmful) responses unrelated to unlearned prompts x_{fgt}, constructed by gathering irrelevant responses from the normal dataset. The three losses in Equation (1) are given by:

$$L_{\text{fgt}} := - \sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_{\text{fgt}}} L(x_{\text{fgt}}, y_{\text{fgt}}; \theta_t),$$

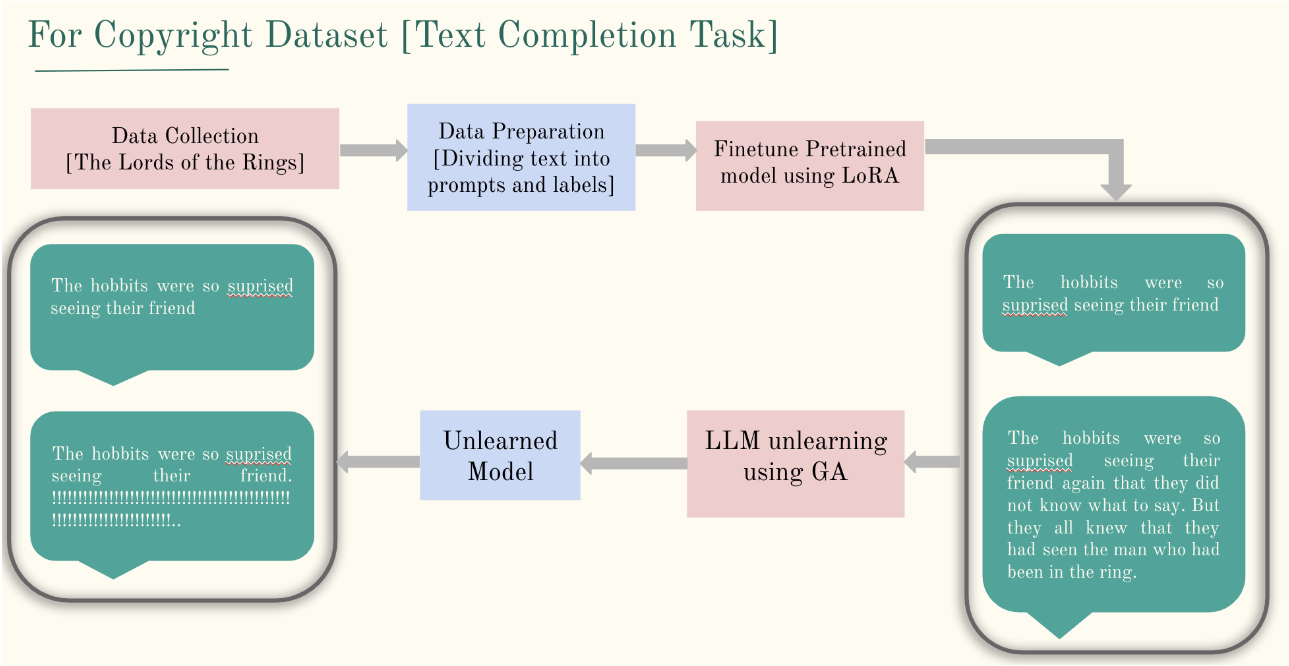

For Copyright Dataset [Text Completion Task]
Data Collection [The Lords of the Rings]
Data Preparation [Dividing text into prompts and labels]
Finetune Pretrained model using LoRA
The hobbits were so suprised seeing their friend
The hobbits were so suprised seeing their friend again that they did not know what to say. But they all knew that they had seen the man who had been in the ring.
LLM unlearning using GA
Unlearned Model
The hobbits were so suprised seeing their friend
The hobbits were so suprised seeing their friend. !!!..

Figure 3.2: Flowchart depicting the unlearning process for Copyright dataset.

$$L_{\mathrm{rdn}} := \sum_{(x_{\mathrm{fgt}},\cdot)\in D_{\mathrm{fgt}}} \frac{1}{|Y_{\mathrm{rdn}}|} \sum_{y_{\mathrm{rdn}}\in Y_{\mathrm{rdn}}} L(x_{\mathrm{fgt}}, y_{\mathrm{rdn}}; \theta_t),$$

L_{fgt} is the gradient ascent loss designed to forget unlearned samples, calculated exclusively on y_{fgt} .

L_{rdn} forces the language model to predict a random output y_{rdn} for the unlearned prompt x_{rdn}, reinforcing forgetting by introducing irrelevance into the predicted outcome. This concept aligns with the idea of label smoothing in classification.

3.1.1 KL LOSS

L_{nor} aims to preserve normal utility by comparing the predicted distribution of the unlearned model with the original language model through forward KL divergence.

$$L_{\mathrm{nor}} := \sum_{(x_{\mathrm{nor}},y_{\mathrm{nor}})\in D_{\mathrm{nor}}} \sum_{i=1}^{|y_{\mathrm{nor}}|} \mathrm{KL}\left(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)\right),$$

where $\mathrm{KL}(\cdot)$ represents the KL divergence term.

3.1.2 JSD LOSS

L_{nor} aims to preserve normal utility by comparing the predicted distribution of the unlearned model with the original language model through Jensen-Shannon Divergence.

$$L_{\mathrm{nor}} := \sum_{(x_{\mathrm{nor}}, y_{\mathrm{nor}}) \in D_{\mathrm{nor}}} \sum_{i=1}^{|y_{\mathrm{nor}}|} \mathrm{JSD}\left(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)\right),$$

where JSD$(\cdot)$ represents the Jensen-Shannon Divergence term.

$$\mathrm{JSD}\left(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)\right) = \frac{1}{2}\mathrm{KL}(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| M) + \frac{1}{2}\mathrm{KL}(h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| M),$$

$$M = \frac{1}{2}(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) + h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i))$$

3.1.3 Hellinger Distance LOSS

L_{nor} aims to preserve normal utility by comparing the predicted distribution of the unlearned model with the original language model through Hellinger Distance.

$$L_{\mathrm{nor}} := \sum_{(x_{\mathrm{nor}}, y_{\mathrm{nor}}) \in D_{\mathrm{nor}}} \sum_{i=1}^{|y_{\mathrm{nor}}|} \mathrm{H}\left(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)\right),$$

where H$(\cdot)$ represents the Hellinger Distance term.

$$H(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i), h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)) = \frac{1}{\sqrt{2}}\sqrt{\sum_{j=1}^{i}\left(\sqrt{h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < j)} - \sqrt{h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < j)}\right)^2}$$

3.1.4 Bhattacharyya Distance LOSS

L_{nor} aims to preserve normal utility by comparing the predicted distribution of the unlearned model with the original language model through Bhattacharyya Distance.

$$L_{\mathrm{nor}} := \sum_{(x_{\mathrm{nor}}, y_{\mathrm{nor}}) \in D_{\mathrm{nor}}} \sum_{i=1}^{|y_{\mathrm{nor}}|} D_{\mathrm{B}}\left(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i) \| h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)\right),$$

where $D_{\mathrm{B}}(\cdot)$ represents the Bhattacharyya Distance term.

$$D_B(h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i), h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < i)) = -\ln\left(\sum_{j=1}^{i}\sqrt{h_\theta(x_{\mathrm{nor}}, y_{\mathrm{nor}} < j)\, h_{\theta_t}(x_{\mathrm{nor}}, y_{\mathrm{nor}} < j)}\right)$$

3.2 Experimental Setup

3.2.1 Datasets

In our experiments, we utilized distinct datasets to train and evaluate the language model. For unlearning harmful responses, we employed the forget dataset (D_{fgt}), specifically PKU-Alignment/PKU-SafeRLHF, containing instances of harmful content. As the normal dataset (D_{nor}) to retain normal behavior during unlearning, we utilized TruthfulQA.

To address the task of unlearning copyrighted content, we curated a custom dataset extracted from the "Lord of the Rings" books. Initially, we fine-tuned our language model on this dataset and subsequently applied our unlearning process. To ensure the preservation of normal behavior, we used the Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books dataset.

For training the text classifier used in our evaluation method, we utilized the toxic comment classification dataset. This dataset is specifically designed for classifying comments based on their toxicity, providing a robust foundation for evaluating the effectiveness of the unlearning process.

3.2.2 Evaluation Method

We propose a evaluation method to measure the effectiveness of unlearned models. Specifically, we train a text classifier on the dataset from which we seek to forget information. Subsequently, we apply our unlearning process to the language model, and we evaluate its performance by testing the output responses using the trained text classifier. This method serves as a quantitative measure for assessing the success of our unlearning approach.

3.2.2.1 Formalization

Let D_{train} denote the training dataset containing information that we aim to forget. We train a text classifier, represented by parameters ϕ, on this dataset. The classifier's accuracy is denoted as $Acc_{\mathrm{classifier}}$.

Next, we employ our unlearning process on a language model, represented by parameters θ, using D_{train}. The updated language model is denoted as $\theta_{\mathrm{unlearned}}$. We then generate responses using $\theta_{\text{unlearned}}$ and evaluate them using the trained text classifier. The accuracy of the classifier on the unlearned model's responses is denoted as $Acc_{\text{unlearned}}$.

3.2.2.2 Effectiveness Metric

The effectiveness of our unlearning process can be quantified using the reduction in the classifier's accuracy when applied to the unlearned model's responses. We define the effectiveness metric E as follows:

$$E = \frac{Acc_{\text{classifier}} - Acc_{\text{unlearned}}}{Acc_{\text{classifier}}} \times 100\%. \tag{3.1}$$

A limitation of this evaluation method is its dependence on the accuracy of the text classifier. The accuracy metric is crucial for determining the model's performance on the specific task of classifying responses. Variations in classifier accuracy may impact the reliability of the effectiveness metric E.

In evaluating the generated content for copyright unlearning within the Lord of the Rings dataset, we employed the BLEU (Bilingual Evaluation Understudy). BLEU is widely used to quantify the similarity between machine-generated text and reference responses. The higher the BLEU score, the closer the alignment between the generated content and the reference responses.

3.3 Results

		Harmful Prompts	**Normal Prompts**
		Harmful Rate (↓)	Similarity to Original
OPT-1.3B	Original	32%	0.659
	Unlearned(KL)	8%	0.403
	Unlearned(JSD)	7%	0.389
	Unlearned(H)	14%	0.303
	Unlearned(D_B)	11%	0.368
OPT-2.7B	Original	41%	0.759
	Unlearned(KL)	11%	0.543
	Unlearned(JSD)	12%	0.551
	Unlearned(H)	19%	0.394
	Unlearned(D_B)	16%	0.485

Table 3.1: Experimental results on unlearning harmful data

The results indicate a significant reduction in the harmful rate after unlearning for both OPT-1.3B and OPT-2.7B. Furthermore, the unlearning process is associated with increased similarity to the original prompts, suggesting that it effectively mitigates the influence of harmful prompts while maintaining the model's alignment with benign inputs. The unlearning process also proves effective in reducing similarity to copyrighted prompts, with minimal impact on similarity to the original ones. This indicates that the unlearning strategy successfully decouples the model from copyrighted data, reducing its association with such prompts while preserving performance on non-copyrighted input.

		Copyrighted Prompts	**Normal Prompts**
		Similarity to Copyrighted	Similarity to Original
OPT-1.3B	Original	0.13	0.611
	Finetuned	0.67	0.102
	Unlearned(KL)	0.01	0.371
	Unlearned(JSD)	0.012	0.341
	Unlearned(H)	0.03	0.275
	Unlearned(D_B)	0.025	0.403
OPT-2.7B	Original	0.27	0.740
	Finetuned	0.71	0.237
	Unlearned(KL)	0.00	0.503
	Unlearned(JSD)	0.00	0.496
	Unlearned(H)	0.03	0.345
	Unlearned(D_B)	0.019	0.438

Table 3.2: Experimental results on unlearning copyrighted data

Overall, the unlearning mechanism appears effective in reducing the influence of harmful and copyrighted prompts on the language model, underscoring its potential to strengthen the model's ethical and legal reliability.

Chapter 4

Extension of Gradient Ascent Unlearning (GAU++)

4.1 Methodology

GA is a proactive way to unlearn D_{fgt}, which takes the inverse update of learning to maximize the model loss on the forget set D_{fgt}. It could be expressed as:

$$\mathcal{L}_{fgt} := - \sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_{\text{fgt}}} L(x_{\text{fgt}}, y_{\text{fgt}}; \theta), \tag{4.1}$$

where x^{fgt} and y^{fgt} respectively correspond to the prompts and responses in the forget data D_{fgt}. L is the loss function, usually defined by Cross-Entropy loss.

$$L_{CE} = -\frac{1}{K} \sum\nolimits_{i=1}^{K} \sum\nolimits_{c=1}^{C} y_{i,c} \cdot log(p_{i,c}), \tag{4.2}$$

where C denotes the size of the vocabulary, and K denotes the total number of tokens in a sequence. $p_{i,c}$ is the probability of token i belonging to class c. $y_{i,c}$ is 1 if the actual token at position i belongs to class c and 0 otherwise. GA forces the model to forget the target data by driving $p_{i,c}$ closer to 0 when $y_{i,c} = 1$.

4.1.1 Issue with GAU (Gradient Explosion)

Since the Cross-Entropy (CE) loss function has no upper bound, adopting GA to unlearn the target information in LLM would increase the gradient without bound and even lead to gradient explosion. To tackle this issue, one naive way is the gradient clipping method, which limits gradient norms with an extra hyper-parameter. Nevertheless, experimental tuning is required to find the optimal hyperparameter. In contrast, we introduce an effective solution that replaces the Cross-Entropy loss with its unlearning version, and uses gradient descent to achieve the unlearning goal. In this way, the issue of gradient explosion could be overcome without the need of tuning extra hyper-parameters.

4.1.2 Mitigating the Issue via Loss Function Reframing

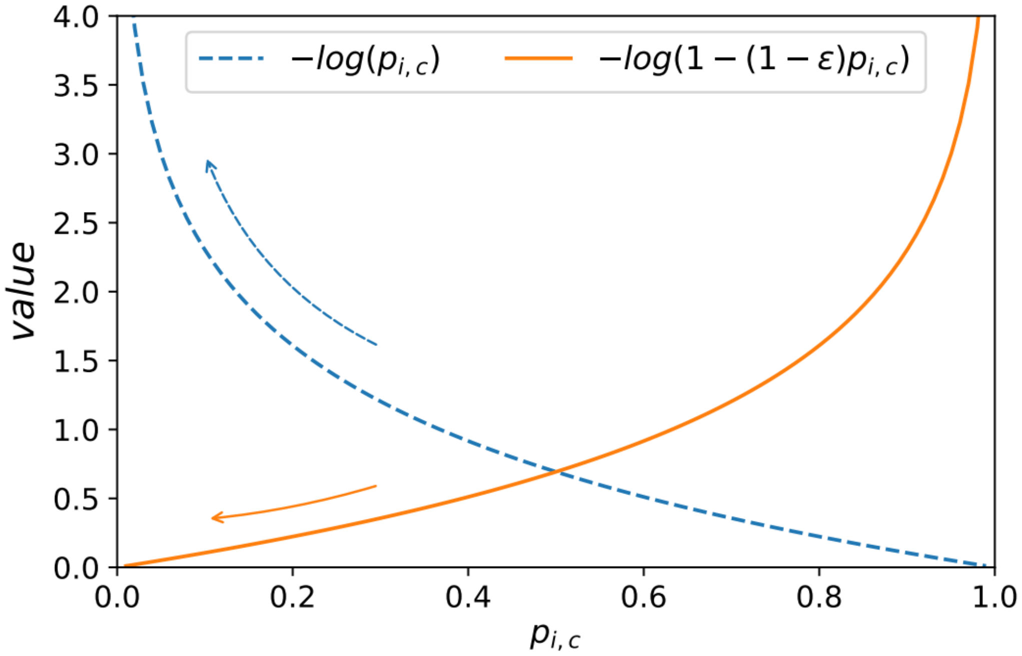


Figure 4.1: When using CE loss and GA for unlearning, pi,c is driven to 0, which leads to gradient explosion (blue line). As for UCE, it adopts the gradientdescent to drive pi,c to 0 to unlearn, preventing gradient explosion.

To tackle this issue, we introduce an effective solution by exploring the previous method of Gradient Ascent Unlearning (GAU), where the update formula is expressed as follows:

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_1 \cdot \nabla_{\theta_t} L_{\text{fgt}} - \epsilon_2 \cdot \nabla_{\theta_t} L_{\text{rdn}} - \epsilon_3 \cdot \nabla_{\theta_t} L_{\text{nor}},$$

where $\epsilon_i \geq 0$ are hyperparameters weighing different losses. however L_{rdn} is same as previous method, Now let's delve into the details of the introduced loss functions L_{fgt}, L_{nor}.

To address this issue, we design the following Unlearning Cross-Entropy (UCE) loss to replace the inverse CE loss (GA):

$$L_{UCE} = -\frac{1}{K}\sum\nolimits_{i=1}^{K}\sum\nolimits_{c=1}^{C} y_{i,c} \cdot log(1-(1-\epsilon)p_{i,c}), \tag{4.3}$$

where ϵ is a small scalar used to slightly scale $p_{i,c}$ to prevent unbounded growth if $p_{i,c} = 1$ in the beginning of unlearning. Following this, $\mathcal{L}_{fgt}$ is defined by Eq. 4.4, enabling model updates via the common gradient descent method. Since the UCE loss has the lower bound 0, it can achieve the goal of unlearning without causing gradient explosion.

$$\mathcal{L}_{fgt} := \sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_{fgt}} L_{UCE}(x_{\text{fgt}}, y_{\text{fgt}}; \theta). \tag{4.4}$$

L_{nor} aims to preserve normal utility by comparing the predicted distribution of the unlearned model with the original language model through forward KL divergence.

$$L_{\text{nor}} := \sum_{(x_{\text{nor}}, y_{\text{nor}}) \in D_{\text{nor}}} \sum_{i=1}^{|y_{\text{nor}}|} \text{KL}\left(h_\theta(x_{\text{nor}}, y_{\text{nor}} < i) \| h_{\theta_t}(x_{\text{nor}}, y_{\text{nor}} < i)\right),$$

where KL$(\cdot)$ represents the KL divergence term.

4.2 Experimental Setup

4.2.1 Models

In conducting our entire unlearning experiments, we employed two distinct language models, OPT-1.3b and OPT-2.7b, Open Pre-trained Transformer Language Models (OPT). These models served as the foundation for investigating the efficacy of our unlearning approach across various scenarios and datasets.

For the text classification task, crucial to our evaluation method, we utilized the BERT (Bidirectional Encoder Representations from Transformers) uncased pretrained model. To adapt the BERT model for our specific classification needs, we fine-tuned it on the toxic comment classification dataset. This allowed us to establish a robust classifier capable of distinguishing toxic and non-toxic content, providing a key component for the evaluation of our unlearning process.

4.2.2 Setup

Our experiments were conducted on the NVIDIA GeForce RTX 3090 GPUs. The experiments were executed for both the OPT-1.3b and OPT-2.7b models.

For training the text classifier, we initiated the process with the toxic comment classification dataset. Utilizing the BERT uncased pretrained model, we conducted training for five epochs, fine-tuned with pretrained weights to enhance the model's ability to discern toxic and non-toxic content.

In the context of unlearning harmful content, as the models were initially prone to generating harmful responses, we directly commenced the unlearning process. Our unlearning algorithm was applied to the forget dataset (D_{fgt}), specifically PKU-Alignment/PKU-SafeRLHF Dai et al. [4], with D_{nor} set as TruthfulQA Lin et al. [8]. The unlearning procedure was executed on pretrained weights for 1000 iterations, utilizing a batch size of 2.

For unlearning copyrighted content, given the model's lack of knowledge about "Lord of the Rings," we initiated the process by fine-tuning the model with a dataset created from the "Lord of the Rings" books. Subsequently, the unlearning algorithm was applied to this fine-tuned model to induce forgetting of "Lord of the Rings" content. To maintain normal behavior, the Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books dataset was employed. This unlearning process was carried out for 1000 iterations with a batch size of 2.

4.3 Results

		Harmful Prompts	**Normal Prompts**
		Harmful Rate (↓)	Similarity to Original
OPT-1.3B	Original	32%	0.659
	GAU	8%	0.403
	GAU++	6%	0.394
OPT-2.7B	Original	41%	0.759
	GAU	11%	0.543
	GAU++	10%	0.549

Table 4.1: Experimental results on unlearning harmful data

The results indicate a substantial reduction in the harmful rate after unlearning for both OPT-1.3B and OPT-2.7B. Additionally, the unlearning process is associated with an increase in the similarity to the original prompts, suggesting that the unlearning mechanism effectively mitigates the influence of harmful prompts while preserving the model's alignment with benign input. The unlearning process demonstrates its effectiveness in reducing the similarity to copyrighted prompts, with a minimal impact on the similarity to original prompts. This suggests that the unlearning strategy successfully disentangles the model from the influence of copyrighted data, contributing to a reduced association with such prompts while maintaining the model's performance on non-copyrighted input.

		Copyrighted Prompts	**Normal Prompts**
		Similarity to Copyrighted	Similarity to Original
OPT-1.3B	Original	0.13	0.611
	Finetuned	0.67	0.102
	GAU	0.01	0.371
	GAU++	0.006	0.389
OPT-2.7B	Original	0.27	0.740
	Finetuned	0.71	0.237
	GAU	0.00	0.503
	GAU++	0.00	0.506

Table 4.2: Experimental results on unlearning copyrighted data

In summary, the unlearning mechanism shows promise in mitigating the impact of harmful and copyrighted prompts on the Language Model, highlighting its potential for enhancing the model's ethical and legal robustness.

Chapter 5

SCRUB Unlearning

5.1 Formulation/Methodology

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).

$$\arg\min_{\mathbf{w}_u} \quad \underbrace{\frac{\alpha}{N_r} \sum_{x_r \in \mathcal{D}_r} d(x_r; \mathbf{w}_u)}_{\text{Stay Close on Retain Set}} + \underbrace{\frac{\gamma}{N_r} \sum_{(x_r, y_r) \in \mathcal{D}_r} L(f(x_r; \mathbf{w}_u), y_r)}_{\text{Cross Entropy on Retain Set}} - \underbrace{\frac{1}{N_f} \sum_{x_f \in \mathcal{D}_f} d(x_f; \mathbf{w}_u)}_{\text{Diverge on Forget Set}}$$

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 $p \in s$.

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

We propose two natural extension of the above SCRUB objective

5.1.1 SCRUB+

We modify the original Scrub objective function by removing the average cross-entropy loss over the retaining set and introducing an additional hyperparameter to better control the forgetting behavior on the forget set. The resulting objective focuses more explicitly on unlearning by emphasizing targeted forgetting through this refined loss formulation.

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

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 to 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.

5.1.2 SCRUB++

We further refine the loss function by introducing a distinct component: the Unlearning Cross-Entropy Loss applied to the forget dataset. To ensure knowledge preservation, we use the average KL Divergence Loss on the retain set, encouraging the model to remain close to its original behavior on retained data. In contrast, both the Unlearning Cross-Entropy Loss and an additional KL Divergence Loss are applied to the forget set to promote divergence from the forgotten information. This carefully balanced objective allows the model to retain useful knowledge while effectively unlearning targeted content. The final loss function is defined as:

$$\arg\min_{\mathbf{w}_u} \underbrace{\frac{\alpha}{N_r}\sum_{x_r\in\mathcal{D}_r} d(x_r;\mathbf{w}_u)}_{\text{Stay Close on Retain Set}} - \underbrace{\frac{1}{N_f}\sum_{x_f\in\mathcal{D}_f} d(x_f;\mathbf{w}_u)}_{\text{Diverge on Forget Set}} + \underbrace{\frac{1}{N_f}\sum_{(x_f,y_f)\in\mathcal{D}_f} L_{UCE}(f(x_f;\mathbf{w}_u),y_f)}_{\text{Diverge on Forget Set}}$$

where L_{UCE} represent the unlearning cross entropy loss function already define in Eq. 4.4.

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

5.2 Experimental Setup

5.2.1 Evaluation Method

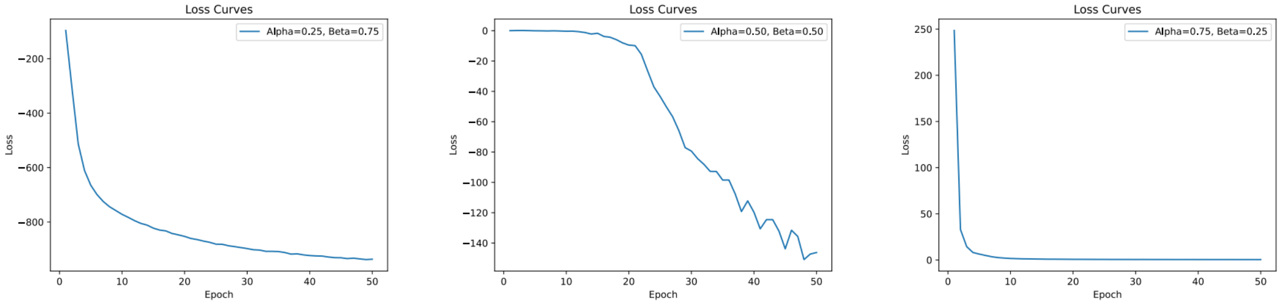
The toxicity metric is an evaluation measurement designed by HuggingFace to assess the level of toxicity present in input texts. Internally, the measure utilizes a pretrained hate speech classification model. We use the default "roberta-hate-speech-dynabench-r4" model, which defines "hate" as "abusive 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.

5.2.2 Experimental details

The experiment aims to adapt the SCRUB algorithm to the domain of text generation in Large Language Models (LLMs). 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.

5.3 Results



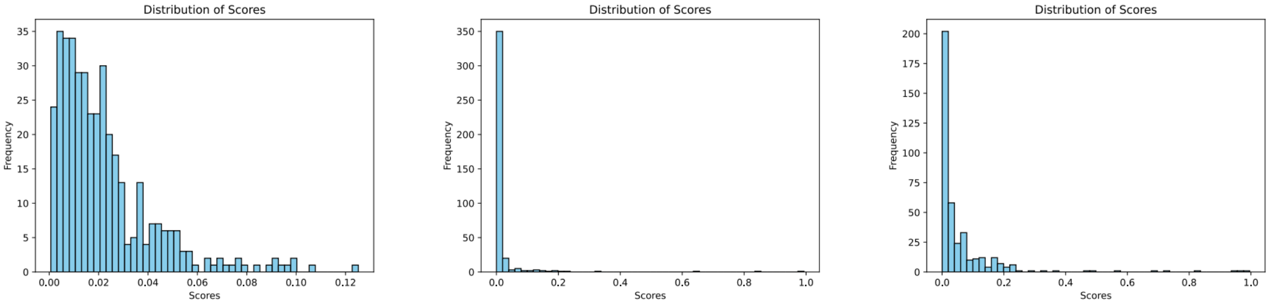
(a) $(0.25, 0.75)$ Loss

(b) $(0.50, 0.50)$ Loss

(c) $(0.75, 0.25)$ Loss

Figure 5.1: Average Loss over Epochs for 3 Models

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) $(0.25, 0.75)$ Toxicity

(b) $(0.50, 0.50)$ Toxicity

(c) $(0.75, 0.25)$ Toxicity

Figure 5.2: 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 original model with pre-trained weights.

Model	**Min Score**	**Max Score**	**Average Score**
Baseline	0.000135	0.979	0.0151
SCRUB+(0.25, 0.75)	0.000639	0.125	0.0610
SCRUB+(0.50, 0.50)	0.000133	0.993	0.0173
SCRUB+(0.75, 0.25)	0.000135	0.996	0.0222
SCRUB++ (0.75)	0.000126	0.985	0.0167

Table 5.1: 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.

5.4 Analysis

The text generations offer insight into the toxicity metric and the different approaches our models took to meet their objective. Looking at Table 5.2, 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

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"	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 5.2: Sample Generations and Toxicity Scores for (0.5, 0.5) Model

"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.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this work, we addressed the critical challenge of machine unlearning in Large Language Models (LLMs), particularly in the context of mitigating harmful and copyrighted content. We proposed and evaluated three complementary methods: Gradient Ascent Unlearning (GAU), GAU++, and a novel Scrub method. The GAU approach, grounded in prior work, was extended by replacing the conventional KL divergence with alternative statistical distances such as Jensen-Shannon Divergence and Bhattacharyya Distance, aiming to enhance the sensitivity of the forgetting process. However, GAU encountered instability due to gradient explosion. To resolve this, GAU++ was introduced with a redesigned loss function that ensured training stability and improved unlearning fidelity. Additionally, the SCRUB+ and SCRUB++ methods offered a fresh paradigm, focusing on structured, selective forgetting with minimal collateral damage to unrelated knowledge—thereby improving safety, controllability, and compliance.

Our results show that it is possible to make large language models forget specific harmful or copyrighted information in an effective and safe way. More than just deleting data, our work shows the importance of unlearning methods that are scalable, understandable, and able to work in real-world systems. This research provides a strong foundation for future work in creating language models that can not only learn new things but also reliably forget when needed—helping ensure safety, fairness, and trust in how these models are used.

6.2 Future Work

The exploration of unlearning in Large Language Models (LLMs) remains a rapidly evolving domain with significant opportunities for advancement. One promising direction is the integration of Reinforcement Learning from Human Feedback (RLHF) into the unlearning pipeline. While

RLHF has shown success in aligning LLMs with human preferences, it can also be extended to penalize the retention of unwanted knowledge—enabling reinforcement-driven unlearning. This would allow models to be guided toward safe, ethical outputs through iterative human feedback that actively discourages harmful, biased, or copyrighted content.

Another critical direction involves influence-based and certifiable unlearning methods. Influence estimation techniques such as TracIn, DFIN, and Fisher Information-based methods can be adapted to LLMs to trace and attenuate the effect of specific training samples. These methods offer a pathway toward more targeted, explainable unlearning with lower computational cost. Complementary to this is the need for formal guarantees of forgetting, using certifiable unlearning frameworks that apply principles from differential privacy and information theory to assure users and regulators that data traces have been effectively removed. Additionally, parameter-efficient unlearning approaches, such as LoRA- or Adapter-based architectures, offer a compelling direction, enabling selective unlearning in subspaces of the model without compromising overall performance.

Looking ahead, expanding unlearning capabilities to multimodal foundation models (e.g., combining text, vision, and speech) represents an essential step toward holistic compliance in real-world systems. Similarly, combining our proposed methods (GAU++, Scrub) with continual learning frameworks can enable models to adaptively learn and forget in response to policy updates or user requests. Finally, the field would benefit from the creation of standardized unlearning benchmarks for LLMs, encompassing metrics for forgetting efficacy, utility preservation, safety compliance, and computational efficiency. Establishing such benchmarks would support robust evaluation and foster the development of practical, trustworthy unlearning systems at scale.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 5

[2] 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. 7, 8

[3] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms, 2023. URL `https://arxiv.org/abs/2310.20150`. 2

[4] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023. 22

[5] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023. 9

[6] Fadi Hassan, David Sánchez, Jordi Soria-Comas, and Josep Domingo-Ferrer. Automatic anonymization of textual documents: detecting sensitive information via word embeddings. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 358–365. IEEE, 2019. 5

[7] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022. 7, 8, 9

[8] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *CoRR*, abs/2109.07958, 2021. URL https://arxiv.org/abs/2109.07958. 22

[9] Zhe Liu and Ozlem Kalinli. Forgetting private textual sequences in language models via leave-one-out ensemble. *arXiv preprint arXiv:2309.16082*, 2023. 8

[10] Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022. 10

[11] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022. 4

[12] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018. 5

[13] Youyang Qu, Xin Yuan, Ming Ding, Wei Ni, Thierry Rakotoarivelo, and David Smith. Learn to unlearn: A survey on machine unlearning. *arXiv preprint arXiv:2305.07512*, 2023. 4

[14] David Sánchez and Montserrat Batet. C-sanitized: A privacy model for document redaction and sanitization. *Journal of the Association for Information Science and Technology*, 67 (1):148–163, 2016. 5

[15] Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A survey and taxonomy. *arXiv preprint arXiv:2305.06360*, 2023. 4

[16] Victoria Smith, Ali Shahin Shamsabadi, Carolyn Ashurst, and Adrian Weller. Identifying and mitigating privacy risks stemming from language models: A survey. *arXiv preprint arXiv:2310.01424*, 2023. 9

[17] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022. 9

[18] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, and Guillem Cucurull. Llama 2: Open foundation and fine-tuned chat models, 2023. 5

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[20] Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*, 2023. 10

[21] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36, 2023. 4

[22] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023. 2, 9

[23] Dongkeun Yoon, Joel Jang, Sungdong Kim, and Minjoon Seo. Gradient ascent post-training enhances language model generalization. *arXiv preprint arXiv:2306.07052*, 2023. 9

[24] Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *arXiv preprint arXiv:2307.03941*, 2023. 4