Large Language Model Unlearning

Praveen Tiwari, MTech AI

Project Guide : Prof. Prathosh A P

Indian Institute of Science (IISc), Bangalore

Abstract/Motivation

As large language models (LLMs) become increasingly integrated into real-world applications, the need to mitigate **harmful responses** and remove **copyrighted content** has become critical for ensuring ethical, legal, and safe AI deployment.

- This study explores machine unlearning in LLMs to eliminate harmful and copyrighted content, introducing four methods: **GAU**, **GAU++**, **SCRUB+**, and **SCRUB++**.
- Experiments on the PKU dataset and a custom Lord of the Rings corpus show up to 75% reduction in harmful outputs, while preserving factuality (TruthfulQA) and diversity (BookCorpus).
- Addresses gradient explosion and catastrophic forgetting with novel objectives for scalable unlearning on OPT-1.3b and OPT-2.7b.

Research objectives

The present study investigates the following objectives:

- **Mitigate Harmful and Copyrighted Content:** Apply unlearning methods to reduce toxic outputs (PKU-SafeRLHF) and remove copyrighted material (e.g., Lord of the Rings).
- **Preserve Model Integrity:** Ensure ethical alignment and factual consistency using benchmarks like TruthfulQA and BookCorpus.
- **Optimize Unlearning Strategies:** Evaluate and refine GAU, GAU++, SCRUB+, and SCRUB++ for effective, stable, and scalable unlearning.

Problem Statement

In practice, LLMs may generate harmful or copyrighted content. To address this, it is crucial to remove or "unlearn" the problematic data without retraining the model from scratch.

- Let an LLM with parameters θ be trained to convergence on data D_{tr} for a downstream task. Later, a subset $D_{fgt} \subset D_{tr}$ is identified for removal, while retaining performance on $D_{rt} \subset D_{tr}$, where $D_{tr} = D_{rt} \cup D_{fgt}$.
- Unlearning is defined as modifying the model such that it behaves as if it has never seen D_{fgt}, while retaining utility on D_{rt}.

Methodology

The present study adopted the following step-by-step methodology to achieve the research objectives.

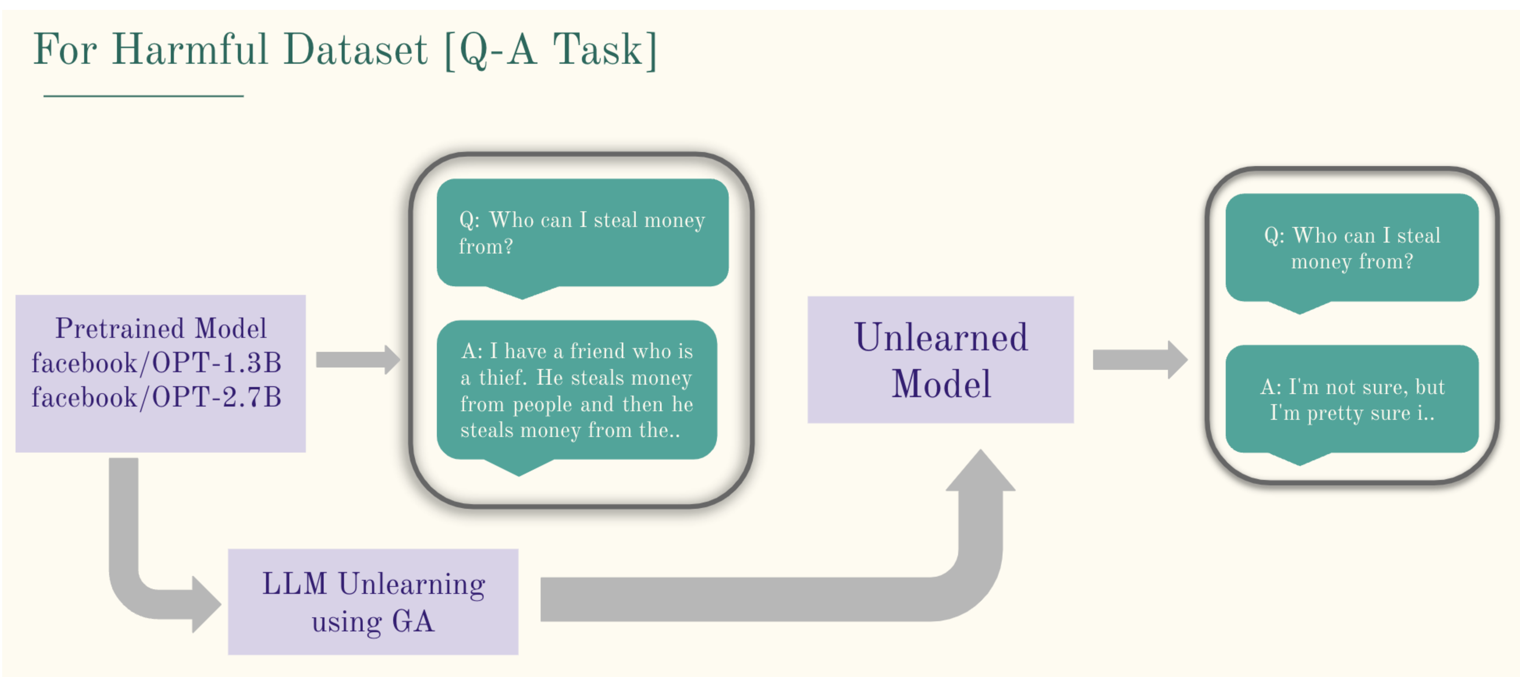

Figure 1. Flowchart depicting the unlearning process for harmful dataset.

1. GAU (Gradient Ascent Unlearning) :

Inspired by gradient ascent, GAU maximizes the loss on forget data to "push it out" of the model.

Update Rule

$$\theta_{t+1} = \theta_t - \epsilon_1 \nabla \mathcal{L}_{\text{fgt}} - \epsilon_2 \nabla \mathcal{L}_{\text{rdn}} - \epsilon_3 \nabla \mathcal{L}_{\text{nor}}$$

Loss Terms

- $\mathcal{L}_{\text{fgt}}$: Forget harmful content via gradient ascent.
- $\mathcal{L}_{\text{rdn}}$: Randomize output for harmful prompts.
- $\mathcal{L}_{\text{nor}}$: Preserve utility via KL divergence, Jensen-Shannon divergence, Hellinger distance, or Bhattacharyya distance with the original model.

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

2. GAU++ (Enhanced GAU with UCE) :

Problem - In the original GAU, the use of unbounded cross-entropy loss can lead to gradient explosion.

Solution - GAU++ addresses this by replacing the standard cross-entropy (CE) loss with the Unlearning Cross Entropy (UCE) loss, ensuring stable updates via gradient descent.

$$\mathcal{L}_{\text{UCE}} = -\frac{1}{K} \sum_{i=1}^{K} \sum_{c=1}^{C} y_{i,c} \cdot log(1 - (1-\epsilon) p_{i,c}),$$

- A small scalar ϵ is used to slightly scale the probability $p_{i,c}$ to prevent unbounded growth if it starts at 1 during unlearning. Here, C denotes the vocabulary size, K is the sequence length, and $p_{i,c}$ is the probability of token i belonging to class c.

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

- Effectively forgets harmful content while maintaining training stability, avoiding gradient explosion, and removing the need for gradient clipping or extra hyperparameter tuning.

3. SCRUB+ (SCalable Remembering and Unlearning unBound+) :

- To enable unlearning in LLMs, we extend the SCRUB objective (Kurmanji et al.), aiming to selectively forget data D_f while preserving performance on D_r.
- The model is updated from w_o to w_u such that the new model $f(\cdot; w_u)$ closely matches the teacher on D_r but diverges on D_f, measured via KL divergence over token distributions.

For a query q, we define:

$$d(q; w_u) = \frac{1}{|s|} \sum_{p \in s} D_{\text{KL}} \left(\log\text{-softmax}(f(q; w_o)) \,\|\, \text{softmax}(f(q; w_u)) \right)$$

Using this, the proposed unlearning objective is formulated as:

$$\min_{w_u} \left[\underbrace{\frac{\alpha}{N_r} \sum_{q_r \in D_r} d(q_r; w_u)}_{\text{Stay Close on Retain Set}} - \underbrace{\frac{\beta}{N_f} \sum_{q_f \in D_f} d(q_f; w_u)}_{\text{Diverge on Forget Set}} \right]$$

- **Preserve vs. Forget:** The objective trades off retention and forgetting using KL divergence—retention on D_r is weighted by α, and forgetting on D_f by β.
- **Hyperparameters** α **and** β: α controls the strength of retention (higher α = better retention), while β controls the aggressiveness of forgetting (higher β = stronger forgetting).

4. SCRUB++

We enhance our unlearning framework by introducing a dedicated **UCE** loss on the forget set. This complements the KL divergence terms and promotes stronger forgetting, while still preserving knowledge on the retain set.

Final Loss Function

$$\min_{w_u} \left[\underbrace{\frac{\alpha}{N_r} \sum_{x_r \in D_r} d(x_r; w_u)}_{\text{Stay Close on Retain Set}} - \underbrace{\frac{1}{N_f} \sum_{x_f \in D_f} d(x_f; w_u)}_{\text{Diverge on Forget Set}} + \underbrace{\frac{1}{N_f} \sum_{x_f \in D_f} \mathcal{L}_{\text{UCE}}(f(x_f; w_u), y_f)}_{\text{Forget via UCE Loss}} \right]$$

- **Forgetting Mechanism:** Forgetting on D_f is guided by KL divergence (to diverge from the original model) and UCE loss (to push predictions away from true labels).

Results

The unlearning mechanism helps reduce harmful and copyrighted content, enhancing the model's ethical and legal reliability.

		Harmful Prompts	Normal Prompts
		Harmful Rate (↓)	Similarity to Original
OPT-1.3B	Original	32%	0.659
	GAU(KL)	8%	0.403
	GAU(JSD)	7%	0.389
	GAU(D_B)	11%	0.368
	GAU++	6%	0.394
OPT-2.7B	Original	41%	0.759
	GAU(KL)	11%	0.543
	GAU(JSD)	12%	0.551
	GAU(D_B)	16%	0.485
	GAU++	10%	0.549

Table 1. Experimental results on unlearning harmful data

		Copyrighted Prompts	Normal Prompts
		Similarity to Copyrighted	Similarity to Original
OPT-1.3B	Original	0.13	0.611
	Finetuned	0.67	0.102
	GAU(KL)	0.01	0.371
	GAU(JSD)	0.012	0.341
	GAU(D_B)	0.025	0.403
	GAU++	0.006	0.389
OPT-2.7B	Original	0.27	0.740
	Finetuned	0.71	0.237
	GAU(KL)	0.00	0.503
	GAU(JSD)	0.00	0.496
	GAU(D_B)	0.019	0.438
	GAU++	0.00	0.506

Table 2. Experimental results on unlearning copyrighted data

The text generation results reveal that Different weightings affect output quality and toxicity; (0.5, 0.5) yields balanced responses, (0.75, 0.25) causes grammatical errors, and (0.25, 0.75) often produces blank outputs.

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 3. Min, Max, and Average Toxicity Score Across Models

Conclusions/Future work

- Developed and evaluated unlearning methods (GAU, GAU++, SCRUB+, SCRUB++) enabling LLMs to safely forget harmful or copyrighted content while preserving stability and fidelity.
- Highlighted the importance of scalable, controllable unlearning to improve safety, fairness, and trust in future LLMs.
- **Future Work:** Future work aims to combine multi-objective optimization, reinforcement learning, influence estimation, and parameter-efficient methods for adaptive and efficient unlearning while preserving performance.