Howard University-AI4PC at SemEval-2025 Task 4: Unlearning Sensitive Content From Large Language Models Using Finetuning and Distillation for Selective Knowledge Removal

Aayush Acharya and Saurav K. Aryal
EECS, Howard University
Washington, DC 20059, USA
https://howard.edu/
saurav.aryal@howard.edu

Abstract

This paper presents our approach and submission to the SemEval 2025 task on "Unlearning Sensitive Content from Large Language Models." The task focuses on making LLMs forget specific knowledge, such as copyrighted material and personally identifiable information (PII), without needing expensive retraining from scratch. We propose a method to unlearn using fine-tuning and knowledge distillation. Our approach involves fine-tuning separate models on "retain" and "forget" datasets to preserve or suppress knowledge selectively. We then distill the model to try to suppress the data using a combine loss of $L2$, KL divergence and $cosine$ similarity while retaining knowledge from the fine-tuned model using KL divergence loss.

1 Introduction

Large language models (LLMs) have revolutionized natural language processing (NLP) by training on a vast amount of publicly available information to perform various tasks. Their ability to generate human-like responses and understand complex linguistic patterns has made them indispensable across industries. During training, they may inadvertently retain copyrighted content or personally identifiable information (PII), leading to ethical, legal, and privacy concerns. The presence of such sensitive data within an LLM raises questions about data security, user privacy, and intellectual property rights, making it crucial to develop mechanisms to selectively remove unwanted information from these models.

To foster research in the field of LLM unlearning, Ramakrishna et al. (2025) introduced SemEval-2025 Task 4: Unlearning Sensitive Contents from Large-Language Models. This task was divided into three sub-tasks:

- Unlearning long-form synthetic documents.
- Unlearning short-form synthetic biographies containing PII such as names, phone number, email, address, Social Security Number(SSN).
- Unlearning real documents sampled from the target model's training data.

The participants were provided with two sets of data: "Forget" and "Retain". As the names suggest, the goal was to ensure the model forgets the information from the "Forget" set and preserves the information from the "Retain" set. Aditionally, the organizers provided 1B and 7B parameter models pretrained on the datasets provided to facilitate the experimentation. We refer to these models as candidate models throughout this work. For the evaluation phase, participants were required to submit working Python scripts implementing the solution to the problem. The scripts were executed on privately held subsets of "Forget" and "Retain" sets for each sub-tasks to assess the effectiveness of the method.

One possible solution to solve the problem is to retrain the model from scratch using a filtered dataset, but this process is computationally expensive and requires substantial GPU hours, making it impractical for large-scale models. Moreover, complete retraining does not guarantee the complete removal of unwanted knowledge as residual traces of the information may persist due to the complex nature of LLMs.

To address these challenges, we experimented with a method to unlearn specific data from LLMs without requiring complete retraining. Our method focuses on selectively modifying the model in a way to suppress unwanted knowledge while preserving required information. We try to achieve this by fine-tuning separate models on forget and retain sets. This allows us to distill knowledge in a way where we try to suppress the forget-set information and retain the retain-set information.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1772–1776
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

2 Related Work

In this section, we explore prior research done in the field of unlearning in non-LLM and LLM settings.

2.1 Non-LLM settings

Ullah et al. (2021) uses gradient manipulation technique to unlearn. The paper introduces Total Variation(TV) stability as a framework for achieving unlearning. The paper proposes a TV-stable algorithm using noisy Stochastic Gradient Descent(SGD).

Setlur et al. (2022) introduces an adversarial unlearning approach to unlearn. The paper introduces Reducing Confidence along Adversarial Directions (RACD) and provides a theoretical analysis to support its effectiveness in unlearning patterns from the training data.

Fan et al. (2024) introduces parameter saliency identification for targeted knowledge removal in image generation. The paper introduces Saliency Unlearning(SalUn) which can achieve up to 100% accuracy when it comes to unlearning.

While prior methods like TV-stable SGD, RCAD, and SalUn have shown their effectiveness, they primarily target smaller-scale models or specific domains like classification and image generation. These approaches may not directly generalize to LLMs. Our approach, however, is specifically geared towards LLMs by utilizing finetuning and knowledge distillation.

2.2 LLM settings

Jang et al. (2023) introduce gradient ascent on the target token to forget information. They also highlight that sequential unlearning is better than batch unlearning. In contrast, our method involves reinforcing the knowledge to eventually retain or forget the information.

Ji et al. (2024) introduces a method that uses logit differences to suppress certain information. They use assistant models to forget the "retain" set and remember the "forget" set and eventually use the logit difference to remove forget set information from the base model. In our method, however, we use assistant models to retain both the retain and forget sets and suppress the forget set onto the candidate model while reinforcing the information from the retain set.

Kassem et al. (2023) introduces DeMem, an approach that utilizes a reinforcement learning feedback loop to unlearn. In contrast, our approach uses finetuning and knowledge distillation to unlearn.

3 Description of Our Technique

This section outlines the methodology used in our experiments.

Before detailing the technique, we establish the following notations:

- Let X represent the dataset on which the candidate LLM was initially trained.
- Let Y represent the unlearning dataset, which contains the information that has to be removed from the candidate model. Here, $Y \subset X$.
- Let Z represent the retain dataset, which contains the information that has to be retained by the candidate model. Here, $Z \subset X$.

The experiment consists of the following steps:

3.1 Fine tuning Models

In the initial phase of our approach, we focused on fine-tuning two distinct candidate models to ensure a clear separation between the information we intended to forget and that which we aimed to retain. In order to make fine-tuning less resource intensive, we quantized the models. As (Lang et al., 2024) suggest, quantization reduces the size of an LLM as we reduce the precision of the model. With the reduction in precision, we get a speedup in mathematical operations like matrix multiplication. This speedup in mathematical operations would reduce the time required for finetuning, however it comes with the cost of reduction in performance of the model.

The first candidate model was fine-tuned in the data set Y, which contains the forget-set data that we specifically seek to remove from the LLM. This fine-tuning process was essential in strengthening the model's knowledge of dataset Y, which would enable us to later suppress this information during the distillation phase. We refer to this fine-tuned model as $Model_{forget}$, as it represents the model that recognizes and remembers the data we wish to unlearn. To achieve effective fine-tuning, we utilized carefully selected hyperparameters tailored to optimize the model's performance on the forget set.

In parallel, we implemented a similar fine-tuning process on another candidate model using dataset

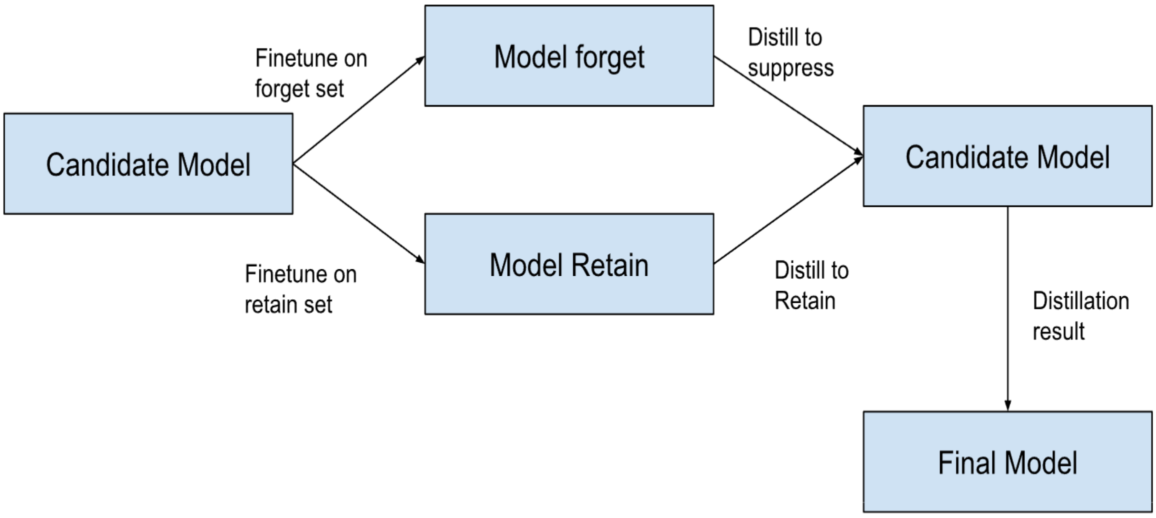

Figure 1: System implementation diagram

Z, which comprises the retain set— the information we wish to preserve and protect from being lost during the unlearning process. This model is referred to as $Model_{retain}$, representing the reinforced understanding of critical knowledge that must remain intact. Just as with $Model_{forget}$, we applied a set of appropriate hyperparameters to ensure optimal performance.

3.2 Knowledge Distillation

Following the fine-tuning of $Model_{forget}$ and $Model_{retain}$, we moved to the phase of knowledge distillation, where we aimed to transfer the insights from these fine-tuned models onto the candidate model. This process required careful attention to ensure that we achieved selective unlearning of the unwanted information while retaining essential knowledge. During the distillation of $Model_{forget}$ onto the candidate model, we implemented a negation mechanism aimed at the logits generated by the teacher model (i.e., $Model_{forget}$). This mechanism attempted to discourage the base model from aligning its outputs with the learned representations from the forget set. To discourage the alignment, we negated the result from Kullback–Leibler divergence(KL divergence), cosine similarity, and L2 loss. The combined loss function is expressed as follows:

Before expressing the loss function, we establish the following notation:

- l_2 refers to l_2 loss.
- KL refers to KL divergence loss.
- $cosine$ refers to cosine similarity.
- α is an arbitrary constant.

$loss = Mean(-\alpha l_2, -KL, -\alpha cosine\ )$

During the training process, the negatively weighted components in the loss function serves to encourage divergence rather than convergence between the predicted and actual values. Specifically, the negatively weighted KL divergence tries to push the predicted probability distribution to differ as much as possible from the true distribution. Similarly, the negatively weighted $cosine$ similarity tries to encourage dissimilarity in vector direction. Likewise, the negative weighted l_2 loss tries to maximize the distance between the predicted and actual output.

Conversely, when distilling knowledge from $Model_{retain}$ onto the same candidate model, we adopted a contrasting strategy. Our goal in this case was to ensure that the logits were closely aligned with those generated by $Model_{retain}$. This alignment attempted to preserve the information within the dataset Z. The loss function that we used to continue to retain the information was KL divergence loss. With KL divergence loss, we tried to bring the probability distribution of the actual and the predicted result as close as possible.

4 Result

Our approach resulted in insufficient GPU VRAM on the official test environment for the 7B parameter OLMo model. However, there was no such issue when it came to running the algorithm on the 1B parameter OLMo model. The final 1B parameter model underperformed and ranked 24th on the leaderboard. Our final aggregate score was 0.079 with an MMLU score of 0.236.

We conducted additional experiments using the GPT-2 model with 137 million parameters. The performance of the model was evaluated using Rogue score. The scores after finetuning the models on Retain and Forget sets are as follows:

	Forget Finetuned	Retain Finetuned
Rogue-1	0.8	0.83
Rogue-L	0.8	0.83

Table 1: Rouge scores after finetuning

The results above indicate that the finetuned models retained some information from their respective training datasets. After the distillation process as discussed in section 3.2, we had the following results:

	Forget dataset	Retain dataset
Rogue-1	0.71	0.77
Rogue-L	0.71	0.77

Table 2: Comparison of Rogue-1 and Rogue-L scores between forget and retain datasets

The post-distillation results suggest that for the retain dataset, the model's output tried to move as close as possible to the actual output. While the result from forget dataset suggest that the predicted outputs shifted a bit from the actual outputs. While the negative weighted loss functions helped in somewhat suppressing the information, we still can see that the suppression was incomplete because of relatively high rogue scores. This finding suggests a potential need for an alternative or complementary loss function to enhance the effectiveness of unlearning.

5 Conclusion

In this study, we explored a fine-tuning and distillation-based approach to unlearning in large language models. Our method involved training separate models to distinguish between forget and retain sets, followed by a distillation phase to suppress unwanted knowledge while preserving critical information. Despite the effort, the results did not achieve the effective unlearning. The method struggled to fully remove the targeted knowledge and resulted in an unintended degradation to the model, highlighting the need for a modified approach to solve the problem.

6 Limitations & Future Work

Our approach of unlearning had limitations that hindered overall effectiveness. Although the negative-weighted loss function contributed to partial suppression of the forget-set, it also affected the model's effectiveness. Our method lacked a rigorous mechanism to confirm whether the targeted information was properly removed.

For future work, we propose several directions. Applying our method on larger models, may reveal whether the increase in model capacity lead to more effective suppression. Additionally, using a non-quantized model during the process could offer higher precision during training, potentially improving the outcomes. Finally, exploring alternative loss functions to suppress the knowledge may result in more robust unlearning behavior.

Acknowledgement

This research project was supported in part by the Office of Naval Research grant N00014-22-1-2714. The work is solely the responsibility of the authors and does not necessarily represent the official view of the Office of Naval Research.

References

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2024. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Preprint*, arXiv:2406.08607.

Aly Kassem, Omar Mahmoud, and Sherif Saad. 2023. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379, Singapore. Association for Computational Linguistics.

Jiedong Lang, Zhehao Guo, and Shuyu Huang. 2024. A comprehensive study on quantization techniques for large language models. *Preprint*, arXiv:2411.02530.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025. Semeval-2025 task 4: Unlearning sensitive content from large language models. *arXiv preprint*.

Amrith Setlur, Benjamin Eysenbach, Virginia Smith, and Sergey Levine. 2022. Adversarial unlearning: Reducing confidence along adversarial directions. In *Advances in Neural Information Processing Systems*.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. 2021. Machine unlearning via algorithmic stability. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4126–4142. PMLR.