DUTir at SemEval-2025 Task 4: Optimized Fine-Tuning of Linear Layers for Balanced Knowledge Forgetting and Retention

Zekun Wang, Jingjie Zeng, Yingxu Li, Liang Yang*, Hongfei Lin
School of Computer Science and Technology, Dalian University of Technology, China
{zk_wang,jjtail,liyx}@mail.dlut.edu.cn
{liang,hflin}@dlut.edu.cn

Abstract

This paper describes our system used in SemEval-2025 Task 4: Unlearning sensitive content from Large Language Models. In this work, we propose a method for controlling the fine-tuning of a model's linear layers, referred to as CTL-Finetune (Control-Tuned Linear Fine-tuning). The goal of our method is to allow the model to forget specific information while preserving the knowledge it needs to retain. The method consists of four main components: 1) shuffling data labels, 2) shuffling label gradient calculation, 3) determination of control layers, and 4) fine-tuning using a combination of gradient ascent and gradient descent. Experimental results demonstrate that our approach effectively enables the model to forget targeted knowledge while minimizing the impact on retained information, thus maintaining the model's overall performance.

1 Introduction

Large Language Models (LLMs) have achieved significant advancements in understanding and solving natural language tasks. However, during the training process, LLMs tend to memorize vast amounts of data (Liang et al., 2022; Ouyang et al., 2022), which may lead to the reproduction of creative content or private information. This, in turn, poses legal risks to model developers and suppliers. These issues are typically identified post model training during testing or red teaming. Moreover, stakeholders may request the removal of their data from the model to protect copyright or exercise their right to be forgotten. However, retraining the model after each data deletion request is prohibitively costly and unsustainable. In light of these challenges, Anil Ramakrishna et al. introduced Task 4, named "Unlearning Sensitive Content from Large Language Models," in SemEval 2025 (Ramakrishna et al., 2025a,b). This task aims to develop a comprehensive evaluation framework to effectively eliminate sensitive data from LLMs, thereby providing a novel solution for the application of unlearning techniques in the domain of LLMs.

The three subtasks are designed with different types of textual data to evaluate the model's "forgetting" capability when handling sensitive information. These include: long-form synthetic creative documents (covering multiple genres), short synthetic biographies containing Personally Identifiable Information (PII) such as fictional names, SSNs, and home addresses, as well as real documents sampled from the target model's training dataset. These tasks aim to comprehensively test the model's ability to identify and eliminate sensitive content across various scenarios.

For this task, we employ the fine-tuned OLMo-7B-0724-Instruct-hf model. Building upon this foundation, we propose a method called CTL-Finetune (Controllable Layer Finetuning), which achieves selective (Dai et al., 2021; Tian et al., 2024b; Liu et al., 2024) information forgetting and retention by fine-tuning the model's linear layers. Figure 1 shows the overview of our framework. This approach involves transforming data labels, calculating relevant gradients, identifying control layers, and combining gradient ascent with gradient descent during fine-tuning. This enables the model to erase specific information while preserving essential knowledge.our system ranked 9th in this competition.

2 Background

2.1 Dataset Description

The challenge consists of three distinct subtasks, each focused on different types of documents. Subtask 1 involves long-form synthetic creative documents spanning various genres. Subtask 2 focuses on short-form synthetic biographies containing per-

*Corresponding author

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 522–526

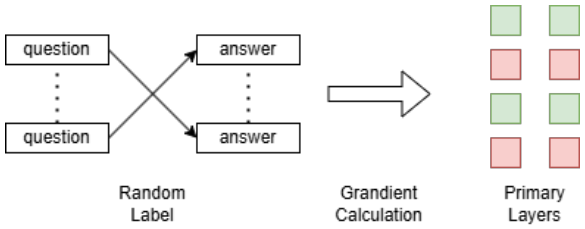

Figure 1: The overview of our framework, we should only unlearn knowledge within the Unlearn Scope while retaining the knowledge within the Retention Scope.

sonally identifiable information (PII), including fake names, phone numbers, Social Security numbers (SSNs), email addresses, and home addresses. Subtask 3 includes real documents sampled from the target model's training dataset.

For each subtask, there are two sets of documents: a Retain set (documents the model should retain in memory) and a Forget set (documents the model should forget). The training set contains 1,110 documents in the Forget set and 1,134 documents in the Retain set, while the validation set consists of 253 Forget documents and 277 Retain documents.

2.2 Related Work

The task of precise knowledge forgetting in neural networks (Yao et al., 2023; Thaker et al., 2024) has gained significant attention, particularly in response to growing concerns around privacy and model security. Several methods have been proposed to allow models to unlearn specific information while retaining essential knowledge. This section reviews key approaches, including gradient ascent-based forgetting, random label fine-tuning, and the use of adversarial examples.

A widely explored approach is gradient ascent, which aims to adjust the model's parameters to weaken its memory of certain knowledge while preserving other information (Jang et al., 2022). This technique explicitly increases the loss associated with specific examples, making the model "forget" particular data points. It is particularly useful for unlearning sensitive or private information without affecting the model's overall performance.

Another method involves random label fine-tuning. In this approach, training data labels are randomly shuffled (Golatkar et al., 2019), creating a disturbance in the model's memory. This disruption helps identify sensitive layers, which can then be fine-tuned to forget specific knowledge while minimizing overfitting. Several studies have shown that randomizing labels effectively aids in forgetting while retaining crucial knowledge.

Adversarial examples have also been explored as a means (Cha et al., 2023) of inducing forgetting. Typically used to test model robustness, adversarial attacks can be leveraged for precision forgetting by generating data points that deliberately disrupt the model's memory. These examples force the model to adjust its parameters, forgetting unwanted information while keeping essential knowledge. However, the challenge lies in balancing the trade-off between forgetting and maintaining performance.

In summary, these methods demonstrate the potential of fine-tuning strategies for precise knowledge forgetting in neural networks. Despite the progress made, challenges remain in effectively controlling the forgetting process, especially in ensuring that models can forget sensitive information without sacrificing their ability to retain useful knowledge. Continued development of targeted techniques, such as gradient ascent, random label manipulation, and adversarial examples, will be critical for advancing this field, particularly in domains requiring high levels of data privacy and security.

3 Methodology

In this work, we propose a fine-tuning approach for the OLMo model, enabling it to selectively forget sensitive knowledge while minimizing the impact on the knowledge that needs to be retained. Our method focuses on fine-tuning specific layers of the model, and it is composed of four main steps. By utilizing this method, we achieve a more nuanced control over the model's memory, allowing it to forget sensitive information without compromising the retention of important knowledge and the model's overall performance. The fine-tuning process ensures that the model adapts to new memory constraints while maintaining its core capabilities.

3.1 Random Shuffling of Dataset Labels

Given that the model has fully memorized the knowledge contained within the documents, we introduce perturbations to the model's memory by constructing new data-label pairs. To achieve this, we randomize the labels in the dataset, which creates a new training set that serves as the foundation for identifying layers of the model that exhibit a significant response to changes in memory. This process of randomization helps to disrupt the model's

established memory, allowing us to evaluate which layers are most influenced by such perturbations.

3.2 Gradient Calculation

Using the shuffled labels, we fine-tune the model on two subsets: the "forget" set and the "retain" set. During fine-tuning, we compute the gradient increments during backpropagation but do not apply these gradients to the model's parameters. This ensures that we capture the gradient information without altering the model's weights. The gradients calculated from the two datasets allow us to analyze which parts of the model are most sensitive to the forgetting and retention processes.

3.3 Identifying Primary Memory Layers

To determine which layers of the model are crucial for memory retention or forgetting, we establish upper and lower bounds for the gradient increments. Layers that fall outside these bounds are excluded from the subsequent gradient update steps. Additionally, we calculate the cosine similarity between the gradient increments from the "forget" and "retain" datasets. If the cosine similarity between these gradients is high for a particular layer(Tian et al., 2024a), it suggests that the layer is simultaneously influencing both the knowledge that should be retained and the knowledge that should be forgotten. Such layers are excluded from further fine-tuning. Only layers with low cosine similarity are retained for updating, ensuring that the model focuses on the layers most relevant for the selective memory process.

3.4 Selective Gradient Update

Once we have identified the layers that are most affected by the forgetting and retention processes, we apply gradient ascent to the layers responsible for forgetting sensitive knowledge. Conversely, for the layers that help retain critical knowledge, we apply gradient descent to preserve this information. This selective application of gradient ascent and descent enables the model to effectively forget sensitive content while safeguarding the knowledge that needs to be maintained.

4 Experimental setup

4.1 Pre-processing

The model and dataset were provided by the task organizers through a Python script that downloads the necessary data and processes it into the appropriate JSON format. For the subsequent model fine-tuning process, we converted the dataset into the standard PyTorch format to facilitate training.

4.2 Dataset Splitting

During the gradient increment computation, all data with shuffled labels were used for fine-tuning based on the Forget and Retain sets. After identifying the model layers to focus on, we randomly selected 50% of the data from the Forget set to perform gradient ascent operations. This approach aimed to minimize the impact on other aspects of the model's performance. For the Retain set, 80% of the data was selected for gradient descent operations, ensuring the model retains the necessary knowledge.

4.3 Evaluation Metrics

The evaluation metrics provided by the task organizers are as follows:

- **Task-specific regurgitation rates**, which were measured using *ROUGE-L* scores on the sentence completion prompts, and the *exact match rate* for question answers, applied to both the Retain and Forget sets. The Forget set metrics were inverted (i.e., $1 -$ metric value) to reflect the model's ability to forget information.
- A **Membership Inference Attack (MIA)** score was computed using a loss-based attack on a sample of member and non-member datasets. The MIA score is given by:

$$1 - |\text{MIA\_loss\_auc\_score} - 0.5| \times 2$$

- Model performance was also evaluated on the **MMLU (Massive Multi-task Language Understanding) benchmark**, which measures test accuracy across 57 STEM subjects.

For the awards leaderboard, only submissions with an MMLU accuracy above 0.371 (which corresponds to 75% of the pre-unlearning checkpoint) are considered. This threshold ensures that the model's utility is not compromised due to unlearning.

Finally, the three scores mentioned above are aggregated to generate a single numeric score for comparing model submissions. The aggregation is done using the **arithmetic mean**.

Algorithm	Final Score	Task Aggregate	MIA Score	MMLU Avg.
Gradient Ascent	0.394	0	0.912	0.269
Gradient Difference	0.243	0	0.382	0.348
KL Minimization	0.395	0	0.916	0.269
Negative Preference Optimization	0.188	0.021	0.080	0.463
CTL-Finetune for 1B	0.172	0.260	0.026	0.229
CTL-Finetune for 7B	0.266	0.205	0.128	0.467

Table 1: Performance Comparison of Various Algorithms

5 Results and Analysis

The final results of our experiments are presented in two models: 7B and 1B, as shown in Table 1. The 7B model achieved a final score of 0.266, which qualifies for the leaderboard in this evaluation. Notably, our model performed well in terms of both the MIA score and the task-aggregate score. When comparing our results with those of other teams, we observed that some teams had MIA and task-aggregate scores of 0, indicating that while our method successfully forgets sensitive information, it also effectively retains the core performance of the model. This highlights the advantage of our approach.

The 1B model, on the other hand, achieved a final score of 0.172. Compared to the 7B model, this result shows a noticeable decline, which could be attributed to the lower structural complexity of the 1B model relative to the 7B model, leading to a reduction in the effectiveness of our method. Overall, while our method demonstrates promising results, there is room for improvement in the forgetting performance, and further optimization is needed to enhance its effectiveness.

6 Conclusion

In this paper, we introduced a novel approach for enabling selective forgetting in large pre-trained models while preserving their core knowledge. Our method, which focuses on fine-tuning specific layers of the model, integrates data shuffling, gradient calculations, and selective updates through gradient ascent and descent. Experimental results demonstrate that our approach can effectively forget sensitive information while maintaining the model's essential performance. The 7B model achieved a final score of 0.266, qualifying for the leaderboard, and showed promising results in terms of MIA and task-aggregate scores, reflecting its ability to balance forgetting and retention. However, the 1B model, with its simpler architecture, exhibited a performance drop, indicating that model complexity plays a significant role in the effectiveness of the forgetting mechanism. Overall, while our method provides a solid foundation for model unlearning, further refinements and optimizations are needed to improve its performance, particularly in terms of the forgetting process.

References

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. 2023. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. *ArXiv*, abs/2301.11578.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *ArXiv*, abs/2104.08696.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2019. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9301–9309.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Towards safer large language models through machine unlearning. *ArXiv*, abs/2402.10058.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and

Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025a. Lume: Llm unlearning with multitask evaluations. *arXiv preprint arXiv:2502.15097*.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025b. Semeval-2025 task 4: Unlearning sensitive content from large language models. *arXiv preprint arXiv:2504.02883*.

Pratiksha Thaker, Yash Maurya, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. *ArXiv*, abs/2403.03329.

Bo Tian, Siyuan Cheng, Xiaozhuan Liang, Ningyu Zhang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. 2024a. Instructedit: Instruction-based knowledge editing for large language models. *ArXiv*, abs/2402.16123.

Bo Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Meng Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024b. To forget or not? towards practical knowledge unlearning for large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Yunzhi Yao, Peng Wang, Bo Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Conference on Empirical Methods in Natural Language Processing*.