GIL-IIMAS UNAM at SemEval-2025 Task 4: LA-Min(E): LLM Unlearning Approaches Under Function Minimizing Evaluation Constraints

Karla Salas-Jimenez[1,2], Francisco López-Ponce,[1,2],
Diego Hernández-Bustamante[3], Gemma Bel-Enguix[1], Helena Gómez-Adorno[3]
[1]Grupo de Ingeniería Lingüística - UNAM
[2]Posgrado en Ciencias e Ingeniería de la Computación - UNAM
[3]Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas - UNAM
{karla_dsj,francisco.lopez.ponce}@ciencias.unam.mx, diegohernandez969@aragon.unam.mx
gbele@iingen.unam.mx, helena.gomez@iimas.unam.mx

Abstract

This paper describes Gradient Ascent and Task Vectors as LLM unlearning methodologies applied to SemEval 2025's task 4. This task focuses on LLM unlearning on specific information under the constraints of preserving the model's advanced text generation capabilities; meaning that our implementations of these algorithms were constrained both in the information datasets as well as the overall effect of each algorithm in the model's general performance. Our implementation produced modified language models that ranked 7th out of 14 valid participants in the 7B parameter model, and 6th out of 24 in the 1B parameter model.

1 Introduction

Large Language Models (LLMs) are one of the most widely used NLP tools for multiple different purposes in and outside of academia and technological research. State of the art models require a substantial amount of training data in order to work at their fullest potential. However, these training datasets may contain personal information and confidential data from multiple sources. If utilized inappropriately, this could result in legal complications arising from infringements of copyright, or the right to be forgotten. Given the probabilistic nature of LLM text generation, and jail-breaking techniques, sensitive information is at risk of being generated in every day usage.

The SemEval 2025 task: Unlearning Sensitive Content from Large Language Models (Ramakrishna et al., 2025) was created to solve this problem. This task works under the assumption that retraining these models from scratch and omitting sensitive information is computationally and economically expensive (Crawford, 2021), meaning that the most efficient approach consists of applying algorithms that modify only certain model weights corresponding to the sensitive information. These modifications should translate to the model being completely unaware of said information, making it unable to generate it by accident, all while preserving the model's text generation capabilities.

Three different evaluation metrics were averaged to obtain the final score: a) the MMLU benchmark average (Hendrycks et al., 2020), in which the unlearned model had to surpass a 0.371 threshold in order to be considered as a valid model, b) a Membership Inference Attack score (Duan et al., 2024), based on attacks sampled from member and nonmember datapoints, and c) task specific regurgitation rates (Lin, 2004), sentence completion measurements focused on forget and retain pieces of information.

In this paper, two unlearning strategies are proposed to solve the problem: 1) Gradient Ascent (Yao et al., 2024) is employed on the data to be forgotten, followed by a fine-tuning step on the data to be retained; and 2) a Task Vector (Ilharco et al., 2023) retraining is implemented to negate the embeddings of the data to be forgotten.

The experiments conducted in this study demonstrate that the efficacy of the unlearning algorithms is influenced by the dimensionality of the model. This task was offered for two models, one with 1 billion (1B) parameters and one with 7 billion (7B) parameters. The 7B model was ranked 7th following the Gradient Ascent implementation, while the 1B model was ranked 6th after the Task Vector implementation. Both rankings are among the 24 teams exhibited in the final evaluation table. The code can be found in GitHub[1].

2 Background

In recent years, the issue of unlearning is being approached from various points of view. A noteworthy work in this area is that of Eldan and Russinovich (2023), which developed a method that enables LLMs to avoid answering with content from

[1]https://github.com/KarlaDSJ/Unlearning-LLM

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1557–1562

input	output
Subtask 1	
Who is the reclusive artist that Shae offered shelter to during the stormy night?	Roz
Who did Catherina seek to protect from Marcile?	The city of Deadesius
Subtask 2	
What is the birth date of Fredericka Amber?	1969-12-21
What is the birth date of Goldi Aqua?	1976-03-29
Subtask 3	
Who is the first woman in Italy to sign a coin, as mentioned in the story?	Laura Cretara
Which poetry collection by Misra won the Sahitya Akademi Award in 1986?	Dwa Suparna

Table 1: Here are some examples of the data for the task. The first example is for the set of information that should be retained. The second example is for the set of information that should be forgotten. For the texts of subtask 1 and 3, a text is given before the questions.

the Harry Potter books by altering the probability of the thematic words.

Articles such as Liu et al. (2024), have made a compilation of works developed in the area to reevaluate the challenges of LLM unlearning, attending to the need of the "right to be forgotten" (Mantelero, 2013). The main categories of challenges have been identified are the following:

1) Defining how to forget. In other words what technical elements are removed: the data or the capacity and functionality of the model. Works in this area would be like the one mentioned in the first paragraph (Eldan and Russinovich, 2023).

2) Influence erasure. This challenge is related to ascertaining the influence of the information to be forgotten on other data. In this domain, various algorithms have been employed to modify the model weights. These include Gradient Ascent and Task Vector, which will be addressed subsequently in this paper. Other algorithms include Gradient Difference, which differentiates between the outcomes of gradient ascent and gradient descent, Relabeling Based Fine Tuning, which involves modifying labels to confuse the model, and others. On the other hand, Patil et al. (2023) has shown that unlearning can be reversed by exploiting the influence they had over other data using extraction or jailbreaking attacks.

3) Unlearning effectiveness is defined as the ability of an algorithm to differentiate between data that should be forgotten and data that should be retained, particularly in cases where this data is interconnected.

4) Efficiency. In this field, the efficiency of the proposed methods for unlearning is studied. Important challenges are presented due to the complexity of LLMs, the infeasibility of pinpointing and attributing training data points designated for unlearning, and the black-box behavior of these models.

Among the aforementioned methods, the contributions of Yao et al. (2024) and Ilharco et al. (2023) are particularly significant. Yao conducted a comparative analysis of Gradient Ascent, Fine-tuning with Random Labels, and Adversarial Samples, concluding that the first method is the most effective. The work of Ilharco addressed modifications to Task Vectors, defined as the weights of a model after adjustment for a specific task. Their findings demonstrated the potential for modifying Task Vectors to facilitate the forgetting of certain information.

To further advance the state of the art in this field, this shared task is proposed (**?**), where there are three subtasks with different types of documents to be forgotten and retained: 1) long-form synthetic creative documents that span a variety of genres, 2) short-form synthetic biographies that contain personally identifiable information (PII), including fictitious names, phone numbers, social security numbers, email addresses, and home addresses, 3) real documents as a sample from the training data set of the target model. Examples of each are shown in Table 1.

3 System overview

The current state-of-the-art was taken into consideration, and two approaches that addressed the problem from different perspectives were selected for testing. The objective was to make a benchmark comparison.

Each approach is characterized by its own sub-

section. The first, Gradient Ascent (GA), examines the problem from the perspective of the loss function, while the second, Task Vector (TV), explores the problem from the viewpoint of the model weights. In this work, we will refer to the set to be forgotten as D_f and the set to be retained as D_r.

3.1 Gradient Ascent (GA)

Yao et al. (2024) mention that applying the gradient ascent on D_f followed by gradient descent on D_r is shown to perform better than other algorithms he tests in his work. So we decided to apply it to this problem.

Gradient Descent (GD) is used to make models learn through minimizing this value. An intuitive idea to forget is to treat the problem in inverse, that is, instead of the model trying to minimize it moves to the opposite side, the Gradient Ascent (GA) can see it as $GA = -GD$. Nevertheless, this could lead to forget other data, not only the D_f. Consequently, a fine-tuning of D_r is necessary to ensure that this information is not missed by the model.

In the initial experiment, the methodology described in the previous paragraph was applied to the text of each instance, as can be seen in Figure 1. Subsequently, it was observed that D_f and D_r exhibited a high degree of similarity in structure, with the only differences being names, numbers, and other identifiers. Since what we want to forget are these identifiers and not the structure of the sentence, a second experiment was performed, passing only the identifiers of each instance in D_f, these identifiers corresponding to a NER tag.

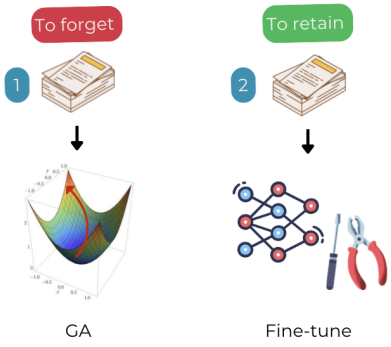

Figure 1: The first method for making a model forget. Gradient ascent with fine-tuning.

3.2 Task Vector (TV)

Another method used is Task Vector. This methodology was proposed by Ilharco et al. (2023) and indicates that a task vector specifies a direction in the weight space of the pre-trained model. That is to say, the direction of these vectors changes when the model improves its performance on the task for which it is being trained. The proposed operations with the task vector are the following: to forget, in which the value of the task vector is negated with the value to be forgotten; to learn, in which the value of the task vector is added to the vector to learn; and to make analogies, in which a combination of the two approaches is utilized.

In order to obtain the task vectors (π_{tv}) for this particular task (t), a fine-tuning (ft) of the given model is performed on D_f. The resulting task vectors are as follows: $\pi_{tv} = \theta^t_{ft} - \theta_{pre}$ Where θ_{pre} is the pretrained weights of the given model and θ^t_{ft} are the weights after fine-tuning on the task t. Therefore, the weights of our new model (θ) are modified according to the following equation: $\theta_{new} = \theta - \pi_{tv}$. The Figure 2 illustrates this process.

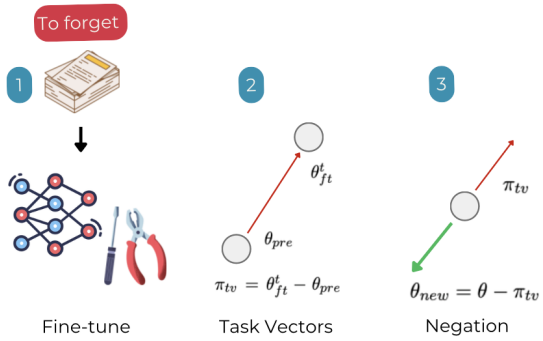

Figure 2: The second method for making a model forget. Task Vector.

For this method, two experiments were conducted. In the first, the model was fine-tuned on the data we wanted to forget without any modifications. In the second, the fine-tuning was performed on the dataset using an NER tag. It is also important to note that in both cases, no layers of the model were frozen.

The results of the application of gradient ascent and task vector can be found in Tables 3 and 4.

4 Experimental setup

All experiments were conducted on the Allenai/OLMo-1B-0724-hf[2] model. The model was quantized to 4-bit using the BitsAndBytesConfig function, and LoRA (Low-Rank

[2]https://huggingface.co/allenai/OLMo-1B-0724-hf

Rank	Team	Final score	Task Aggregate	MIA score	MMLU Avg.
7B					
1	AILS-NTUA	0.706	0.827	0.847	0.443
7	GIL-IIMAS UNAM	0.380	0.478	1.0	0.446
14	ma****8@gmail.com	0.154	0.0	0.0	0.463
1B					
1	AILS-NTUA	0.688	0.964	0.857	0.242
6	GIL-IIMAS UNAM	0.416	0	0.98	0.269
24	ai**c@protonmail.com	0.079	0	0	0.236

Table 2: The highest and lowest scoring teams in the shared task by model are shown, as well as our team score (GIL-IIMAS UNAM)

Adaptation) was employed to reduce resources and enhance efficiency and speed when applying the various approaches. We decided to take the most commonly used parameters to focus on the two proposed methods (GA and TV). This facilitates the comparison of results with the work developed in the state of the art. For LoRA we use the peft package with the following parameters: lora_r = 8, bias='none', lora_alpha = 32 and lora_dropout = 0.0 For FineTune we use the trl package: SFTConfig and SFTTrainer. This makes the task easier.

Three metrics were used to evaluate this task:

MMLU Average: Measured as test accuracy on 57 STEM subjects, models that scored less than 0.371 were discarded to ensure that the model was not extremely damaged by forgetting data, i.e., forgetting much more than it should have.

Membership Inference Attack (MIA) score is a metric that utilizes questions posed to the model to ascertain whether data that should not be present is, or vice versa. In this metric, instances of members and non-members of the model are provided, and the formula $1 - abs(mia\_loss\_auc\_score - 0.5) * 2$ is employed. The optimal MIA rates that can be attained are approximately 0.5. Scores approaching 1 indicate under-unlearning, while scores approaching 0 indicate over-unlearning.

Task Aggregate (TA) is measured using ROUGE-L scores. The ROUGE-L scores indicate the exact match of the answers to the questions in both the set to be forgotten and the set to be retained.

The final score is an average of these three metrics.

5 Results

It can be observed that the efficacy of each approach is model-dependent. For the 1B model, the TaskVector approach yielded superior results (Table 3); however, for the 7B model, it did not perform optimally. This approach was found to be particularly aggressive, as it forgot everything and began answering questions with little to no sense.

This may be because we work with vectors, and these vectors can capture more than just D_f, so negating them greatly affects the performance of the model, this means that the model forgets more things than it should.

Conversely, the most effective approach for the 7B model was found to be gradient ascent, followed by fine tuning (Table 4). Thus, we could get 7th place for the forgetforget7B model and 6th place for the 1B model. As the Table 2 shows. Examples of the output in the 7B model are shown below.

Example of an instance from Gradient ascent with finetuning on 7B model:

Input - Goldi Aqua was born on March 29, 1976. She can be reached via phone at 565-577-9919 and email at goldi_aqua@me.com. Her home

Answer to forget - address is 1011 San Jose Street, Brooklyn Park, MD, 21225. Her Social Security Number is 900-57-8002.

Answer - address is: 730 N. Broadway, San Diego, CA 92101.

Example of an instance from TaskVector on 7B model:

Input - What is Tiffi Magenta's Social Security Number?

Answer to forget - 900514344

Answer - active Cont Cont Cont settings out Cont out le le le le le

Another observation is that the implementation of these approaches exclusively on the NER labels

does not enhance the outcomes; in some cases, it has been observed that this approach can lead to a deterioration in the results, depending on the particular model in use. A subsequent analysis of the MIA score results indicates a tendency towards a value of 1, suggesting a lack of sufficient unlearning.

	Final	TA	MIA	MMLU
GA	0.357	0	0.843	0.229
GA + NER	0.356	0	0.84	0.229
TV	**0.416**	**0**	**0.98**	**0.269**
TV + NER	0.409	0	0.98	0.247

Table 3: Results for the 1B model. "GA" refers to "gradient ascent + finetuning," and "TV" refers to "TaskVector." The result that was reported on the competition page is in bold.

Finally, we can mention that although the algorithms indicate that they work, we can notice that there is an area of improvement and this may be due in part to the fact that both approaches are an aggressive modification of the model. This, along with the fact that the experiments were carried out without freezing any layer, allowing the algorithms to modify the model at different levels and not only in the last layers, could be one of the reasons why the performance was not optimal.

	Final	TA	MIA	MMLU
GA	**0.380**	**0.478**	**1.0**	**0.446**
GA + NER	0.164	0	1.0	0.493
TV	0.399	0	0.475	0.247
TV + NER	0.406	0	0.475	0.269

Table 4: Results for the 7B model. "GA" refers to "gradient ascent + finetuning," and "TV" refers to "TaskVector." The result that was reported on the competition page is in bold.

6 Conclusion

These unlearning methodologies generated notable changes in both versions of the language model, in some cases even more than necessary.

Task Vectors theoretically modify only the weights corresponding to the forget information, yet in this case those weights were highly relevant to the overall capabilities of the model. As a matter of fact, Task Vectors did not manage to obtain non-zero scores in the task aggregate sections of the evaluation, and the model ultimately underperformed in the MMLU, disqualifying that methodology from full evaluation. Freezing certain layers of the model could improve the results both in execution time and in evaluation results.

Gradient Ascent, on the other hand, proved to be the better performing unlearning methodology. To our surprise, the unaltered application of this algorithm performed considerably better than a NER adjusted implementation, suggesting that this algorithm works better with raw data even if the information varies mainly in information that can be tagged with standard NLP methodologies.

In future work, we plan to improve the finetuning parts by adjusting hyperparameters and freezing some layers of the model. Additionally, a more detailed analysis of how the task vectors are created is necessary to ensure that no extra information is removed that affects the model's performance.

Acknowledgments

This paper was supported by UNAM, PAPIIT project IG400325 and IN104424. Karla Salas-Jimenez (CVU: 1291359), and Francisco López-Ponce (CVU: 2045472) thank the CONAHCYT graduate degree scholarship program.

References

Kate Crawford. 2021. *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *Preprint*, arXiv:2402.07841.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *Preprint*, arXiv:2310.02238.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. *Preprint*, arXiv:2212.04089.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summariza-*

tion Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024. Rethinking machine unlearning for large language models. *Preprint*, arXiv:2402.08787.

Alessandro Mantelero. 2013. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *Preprint*, arXiv:2309.17410.

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*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. *Preprint*, arXiv:2402.15159.