Soft Prompting for Unlearning in Large Language Models

Anonymous ACL submission

Abstract

The widespread popularity of Large Language Models (LLMs), partly due to their emerging in-context learning ability, has highlighted the importance of ethical and safety considerations for deployment. Motivated by corresponding data protection guidelines, we investigate machine unlearning for LLMs. In contrast to the growing literature on fine-tuning methods to achieve unlearning, we focus on a comparatively lightweight alternative called soft prompting to realize unlearning in LLMs. With losses designed to enforce forgetting as well as utility preservation, our framework **S**oft **P**rompting for **U**n**l**earning (SPUL) learns prompt tokens that are prepended to a query to induce unlearning of specific training examples at inference time without updating LLM parameters. We conduct a rigorous evaluation of the proposed method, and results indicate that SPUL can significantly improve the trade-off between utility and forgetting for text classification and question-answering. We further validate our method with LLMs of varying parameter sizes to highlight its flexibility and provide detailed insights into the choice of hyperparameters and the influence of the size of unlearning data. Our implementation is available at `https://tinyurl.com/softprompt`.

1 Introduction

With evolving transformer models (Vaswani et al., 2017) and the availability of massive text corpus, language models are progressing rapidly. The *pre-train and fine-tune* pipeline has garnered wide popularity, especially since the release of LLMs such as GPT (OpenAI, 2024) and LLaMA (Touvron et al., 2023). However, several ethical and security concerns have been raised due to the presence of private, sensitive, or harmful information in the training data. For example, LLMs can regurgitate personal information (Nasr et al., 2023), or mimic harmful and/or hateful behavior as a consequence of such content being prevalent in the data (Wen et al., 2023). The non-consented and unwarranted use of copyrighted content for LLM training has also raised significant concerns (Eldan and Russinovich, 2023; Grynbaum and Mac, 2023).

Current policies governing the use and distribution of such models do not encompass all ethical avenues; nonetheless, certain regulations such as California Consumer Privacy Act (CCPA) and GDPR's Right to be Forgotten (RTBF) serve as guidelines for organizations to ensure that their operations do not infringe upon user privacy. Specifically, these regulations stipulate that businesses and data collectors provide and exercise an *opt-out* mechanism essentially allowing individuals to request the deletion of their data on reasonable grounds. In machine learning literature, these regulations have been conceptualized as machine unlearning (Cao and Yang, 2015; Bourtoule et al., 2021), which aims to eliminate the influence of unwanted data points on a model's behavior as if they had never been observed during training. Naturally, machine unlearning should be integrated into the LLM pipeline to address previously outlined issues resulting from the presence of sensitive or harmful data in pre-training. However, unlearning in LLMs faces unique challenges due to the inaccessibility of model and pre-training data, and the sheer size of the pre-trained LLMs making re-training practically infeasible. Much of the research in this direction therefore focuses on the fine-tuning approach which involves training all or a subset of LLM parameters to enforce unlearning (Jang et al., 2023; Chen and Yang, 2023; Yao et al., 2024b; Maini et al., 2024; Yao et al., 2024a).

In contrast, we present a resource-efficient approach to unlearning in LLMs via soft prompting. Soft prompting optimizes a set of task-specific token embeddings that learn contextual information from a corresponding dataset and instruct a frozen LLM during inference (Lester et al., 2021; Li and Liang, 2021). We leverage this ability to mod-

ulate LLM outputs using learnable prompts and formulate **S**oft **P**rompting for **U**n**l**earning (SPUL), a resource-efficient mechanism to achieve LLM unlearning in text classification and multiple-choice question answering (MCQA) tasks. SPUL optimizes soft prompts so that they learn to encode underlying information in the data relevant for unlearning. When prepended to the input tokens of an LLM during inference, the soft prompts guide the LLM towards a *generic response*. We implement a multi-objective loss aligned with specific unlearning goals to facilitate soft prompt learning. SPUL unlearns undesirable outcomes without updating large-scale LLM parameters and can fully capitalize on the language understanding capability offered by the pre-trained LLMs. Consequently, the same pre-trained LLM can be utilized for different unlearning tasks and datasets during inference.

We formulate LLM unlearning as the forgetting of a subset from the training dataset composed of benchmark NLP datasets for sentiment classification or MCQA (Pawelczyk et al., 2024; Li et al., 2024; Chen and Yang, 2023). The unlearning subset could contain data corresponding to some entities or sensitive information that could be used to cause harm. To evaluate whether SPUL can achieve unlearning, we focus on quantifying the predictive performance on the unlearning subset as well as a general retain subset and consider the tradeoff between them. In this setting, SPUL can effectively induce forgetting during inference while preserving the pre-trained utility with significant improvements over multiple baselines that implement fine-tuning. We conduct experiments to analyze the influence of SPUL hyperparameters including the contribution of loss components and the size of the soft prompts. We further validate SPUL on multiple pre-trained LLMs with different parameter sizes and using varied sizes of unlearning sets.

2 Related Work

2.1 Soft prompting

Soft prompting or prompt tuning emerged as a lightweight alternative to fine-tuning while keeping pre-trained LLM parameters frozen. Motivated by discrete prompts that guide pre-trained LLMs via task-specific instructions or demonstration examples, soft prompting makes prompt design more efficient by employing trainable prompt parameters. Lester et al. (2021) added trainable embeddings to the encoder input sequence of an LLM and achieved performance comparable to fine-tuning on NLP classification tasks with models having over 10B parameters. Simultaneously, Li and Liang (2021) developed the notion of prefix tuning which prepends task-specific prefixes to the input embeddings along with the encoder and decoder inputs of an autoregressive LM and obtained comparable performance for text generation tasks. Liu et al. (2021) concatenated trainable continuous prompts with discrete prompts along with a prompt encoder that maps prompts to model inputs to improve performance on supervised and few-shot tasks. Subsequent research showed that deep prompt tuning achieves comparable performance to fine-tuning across several tasks on models of varying scales by inserting tunable parameters into every LLM layer (Liu et al., 2022).

2.2 Unlearning in LLMs

Machine unlearning addresses data protection guidelines by efficiently forgetting training samples corresponding to unlearning requests in place of costly retraining (Bourtoule et al., 2021; Cao and Yang, 2015; Guo et al., 2020; Sekhari et al., 2021) and is also gaining prominence in LLMs due to concerns regarding bias, toxicity, and privacy (Si et al., 2023; Liu et al., 2024). Some works in this direction emphasize model parameter optimization via gradient ascent (Jang et al., 2023; Chen and Yang, 2023; Yao et al., 2024b; Maini et al., 2024; Yao et al., 2024a) to unlearn unwanted responses for specific examples or datasets. They also fine-tune the model with various knowledge alignment objectives to maintain model utility. Other works leverage parameter optimization via relabeling of unlearning data. For instance, Eldan and Russinovich (2023) unlearn Harry Potter content by fine-tuning the model via gradient descent to replace the model's response with outputs containing generic translations. Jia et al. (2024) utilize similar fine-tuning objectives with a focus on optimizer selection and propose a framework that performs influence-based model updates via second-order optimization. Additionally, some works propose localization-based objectives that aim to identify a subset of model units that represent unlearning data and effectively delete them (Meng et al., 2022; Yu et al., 2023; Wu et al., 2023). A few works also focus on modifying LLM input sequences to promote unlearning for black-box LLMs but are limited in the size of unlearning data. For instance, Pawelczyk et al. (2024) formulate in-context unlearning

by crafting inputs comprising unlearning samples paired with flipped labels. Thaker et al. (2024) investigate guardrail techniques for unlearning by instructing models to withhold unwanted knowledge or filtering undesirable LLM outputs. Unlike most fine-tuning-based approaches, our goal in this work is to develop a soft prompting strategy to facilitate unlearning in LLMs. We aim to modulate LLM behavior using prompts similar to input modification strategies. However, instead of specifying manual instructions or providing demonstration samples as context, we leverage soft prompting to automate prompt optimization while adhering to unlearning objectives through loss specifications.

3 Soft Prompting for Unlearning

3.1 Soft Prompting

Let $D = \{s_i, y_i\}_{i=1}^{N}$ denote a dataset containing N input-output pairs where s_i is a text sequence containing n_i tokens and y_i is the corresponding output. Also, let h_θ represent a pre-trained LLM with parameters θ; h_θ can be prompted with s_i to obtain an output $\hat{y}_i$. Assume $\mathbf{x}_i \in \mathbb{R}^{n_i \times d}$ denotes the token embeddings obtained for an arbitrary text sample s_i from the embedding module of h_θ where d is the dimension of the embedding space. We first define p prompt tokens as $\boldsymbol{\phi} = \{\phi_1, \cdots, \phi_p\}$ where $\phi_i \in \mathbb{R}^d$. To adapt h_θ over D using soft prompts, $\boldsymbol{\phi}$ is appended to $\mathbf{x}_i$ to form the sequence $\{\boldsymbol{\phi}, \mathbf{x}_i\} \in \mathbb{R}^{(p+n_i)\times d}$ as input to the encoder or decoder in h_θ. During backpropagation, the pre-trained parameters θ are frozen and gradient updates are applied only to $\boldsymbol{\phi}$ when maximizing the likelihood of the output y_i as

$$\underset{\boldsymbol{\phi}}{\operatorname{argmax}} \log h_\theta(\{\boldsymbol{\phi}, \mathbf{x}_i\}). \tag{1}$$

The size of the learnable prompts $\boldsymbol{\phi}$ is very small compared to that of the pre-trained parameters θ. Nonetheless, soft prompting has shown considerable performance over various language tasks with results comparable to fine-tuning. This motivates us to consider *whether we can achieve unlearning in LLMs by optimizing continuous prompt tokens.*

3.2 Problem Formulation

Given a training dataset D^{tr} that was observed during pre-training of h_θ, we assume a forget set, $D_f^{tr} \subset D^{tr}$, as the data intended for forgetting/removal from h_θ. Simultaneously, we define a retain set $D_r^{tr} = D^{tr} \setminus D_f^{tr}$ comprising the remaining samples. Then, the goal of unlearning is to forget the token sequences in D_f^{tr} while maintaining inference utility on D_r^{tr}. For our work, we focus on the task of text classification and question answering and interpret unlearning as the forgetting of the predictive output token sequences $y_i \in D_f^{tr}$. Essentially, we de-correlate text features and their corresponding labels for the relevant forget samples but preserve the predictive performance on the retain samples. To this end, we aim to design a soft prompting framework to obtain optimized prompt tokens that can guide the base model toward the forget and retain objectives. With our framework, we aim to address the following research questions.

RQ1: How can soft prompting be utilized to effectively unlearn subsets of training data in the text classification/QA domain?

RQ2: How can soft prompting be implemented to achieve utility preservation with forgetting?

RQ3: How efficient is soft prompting-based unlearning compared to fine-tuning?

3.3 Method

As soft prompts can be trained to encode signals from a dataset with the purpose of adapting a pre-trained LLM to a specific downstream task, we anticipate that the strategy can also be utilized to encode relevant information from an unlearning dataset containing forget and retain samples. Here, we propose the framework SPUL that leverages soft prompting to obtain effective prompt tokens $\boldsymbol{\phi}$ from an unlearning dataset D^{tr}. Since one of the unlearning objectives in our framework is to promote feature and text de-correlation for forget samples, we design a loss attuned to enforcing incorrect predictions for the respective text inputs. Specifically, we force the model to associate each input forget text sequence with a generic output token instead of its true label. We construct a generic label set $\bar{Y}$ that is disjoint from the task labels and contains tokens such as *neutral*, *unknown*, or *none* and define a loss over the forget inputs,

$$\mathcal{L}_f = \sum_{(\mathbf{x}_i,.)\in D_f^{tr}} l(\hat{y}_i|\{\boldsymbol{\phi}, \mathbf{x}_i\}, \bar{y}_i), \tag{2}$$

where $\bar{y}_i$ denotes a uniform random sample drawn from the pre-defined generic label set $\bar{Y}$, and $l(\cdot)$ refers to the standard cross-entropy loss. Ideally, $\mathcal{L}_f$ allows the prompt tokens $\boldsymbol{\phi}$ to capture specific nuances from the samples in D_f^{tr} and consequently guide the LLM to change its predictive sequence for an arbitrary example containing the learned distinctions. Simultaneously, unlearning also aims

to preserve the predictive performance for samples not included in the forget set. In SPUL, the prepended prompt tokens ϕ should not change the predictive sequences for $\mathbf{x}_j \in D_r^{tr}$. Therefore, to preserve inference utility on the retain set, we define a loss using their true labels as

$$\mathcal{L}_r = \sum_{(\mathbf{x}_j, y_j) \in D_r^{tr}} l(\hat{y}_j | \{\phi, \mathbf{x}_j\}, y_j), \quad (3)$$

where $l(\cdot)$ again represents the cross-entropy loss. L_r ensures that the model's utility on the retain set does not degrade with the addition of prompt tokens. In addition to maintaining performance on the retain set, the model after unlearning should closely resemble the model before unlearning. In our framework, we constrain the predictive distribution of the model such that $h_\theta(\{\phi, \mathbf{x}_j\})$ reflects $h_\theta(\mathbf{x}_j)$ for any $\mathbf{x}_j \in D_r^{tr}$. We quantify this difference using KL divergence as

$$\mathcal{L}_{kl} = \sum_{(\mathbf{x}_j, .) \in D_r^{tr}} \mathrm{KL}(h_\theta(\{\phi, \mathbf{x}_j\}) || h_\theta(\mathbf{x}_j)), \quad (4)$$

where $\mathrm{KL}(\cdot)$ denotes the KL divergence term. $h_\theta(\{\phi, \mathbf{x}_j\})$ represents the base model's predictive distribution conditioned on inputs prepended with the learnable prompt tokens and $h_\theta(\mathbf{x}_j)$ refers to the output distribution conditioned only on the input text sequence. We utilize $\mathcal{L}_{kl}$ in addition to $\mathcal{L}_r$ to avoid large deviations in the base model's output due to the influence from L_f. Finally, at each time step t during training, we update ϕ by optimizing the overall loss obtained as

$$\mathcal{L} = \mathcal{L}_f + \alpha \cdot \mathcal{L}_r + \beta \cdot \mathcal{L}_{kl}, \quad (5)$$

where α and β are hyperparameters that specify the contribution of the respective loss components.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate SPUL on standard NLP datasets SST-2 (Socher et al., 2013) and Yelp polarity (Zhang et al., 2015) for sentiment classification task. SST-2 and Yelp contain reviews with each text sequence being labeled as a positive or negative sentiment. To build a realistic unlearning scenario where unlearning requests from each user would likely include multiple related training samples, we preprocess the classification datasets to construct the forget and retain sets such that the forget samples are semantically similar to each other (Yelp) or refer to common entities (SST-2). For SST-2, we perform Named Entity Recognition to identify named personalities, select a specific set of entities, and sample all related reviews to form the forget set D_f^{tr}. The remaining reviews are consequently assigned to the retain set D_r^{tr}. We perform a similar partitioning using the selected entities on the test set to obtain D_f^{te} and D_r^{te}. For Yelp, we perform k-means clustering with cosine distance on the training data to divide the reviews into semantically similar groups. We randomly select a subset of the clusters and group them to form the D_f^{tr} and the rest as D_r^{tr}. We utilize the same cluster centers to infer cluster identities for the test data and form the sets D_f^{te} and D_r^{te} accordingly.

Table 1: Dataset Statistics

Dataset	D_f^{tr}	D_r^{tr}	D_f^{te}	D_r^{te}
SST-2	1425	46331	610	19855
Yelp polarity	5081	95012	885	18089
WMDP+SciQ	900	12679	373	1000

We also evaluate SPUL for multiple-choice question answering using a combination of WMDP (Li et al., 2024) and SciQ (Welbl et al., 2017) datasets. SciQ consists of exam questions about Physics, Chemistry, and Biology, among others in a four-way multiple-choice format where each answer choice is associated with symbols such as "A", "B", etc. and WMDP contains questions about hazardous knowledge in biosecurity, cybersecurity, and chemical security in the same format. In this MCQA task, we construct forget sets to unlearn potentially harmful information while retaining general science knowledge, i.e.: we obtain the forget sets from WMDP containing questions about hazardous knowledge in biosecurity and the retain sets from SciQ with general science questions. We refer to this dataset as WMDP+SciQ. Table 1 includes the sizes of the constructed forget and retain sets.

Baselines We assess the effectiveness of SPUL by comparing its performance against multiple SOTA parameter-tuning baselines. Gradient Ascent (GA) (Jang et al., 2023) optimizes pre-trained LLM parameters by maximizing the cross-entropy loss defined only on the forget set D_f^{tr} in place of standard minimization. Fine-tuning with Random Labels (RL) (Golatkar et al., 2020; Yao et al., 2024a) similarly optimizes the LLM on D_f^{tr} but by enforcing convergence on random vocabulary terms. We use the generic label set discussed in Section 3.3 as the random labels for RL. Gradient Ascent + KL Divergence (GA + KL) and Gradient

Ascent + Descent (GA+GD) incorporate parameter optimization on the retain set D_r^{tr} in addition to the GA loss to balance forgetting effectiveness with utility (Yao et al., 2024a). GA + KL defines a KL-divergence constraint on the LLM's output distribution for D_r^{tr} and GA + GD minimizes the standard cross-entropy loss on D_r^{tr}. We fully fine-tune the LLM for all baselines following prior works based on their publicly available implementations.

Settings We use LLaMA-2-7B (Touvron et al., 2023) as the base LLM to evaluate SPUL and further validate its unlearning effectiveness with OPT-1.3B (Zhang et al., 2022) and LLaMA-2-13B (Touvron et al., 2023). To ensure familiarization with the unlearning dataset, we fine-tune the base LLMs on the full training dataset $D^{tr} = D_f^{tr} \cup D_r^{tr}$ for 10 epochs on SST-2, 2 epochs on Yelp, and 5 epochs on WMDP+SciQ with a learning rate set to 0.0001 and context length to 1024 using QLoRA (Dettmers et al., 2023). We treat this fine-tuned version of the LLM as the base model for unlearning. For SPUL, we fix the learning rate at 0.0001 across all LLMs and datasets. We vary prompt token length p among $\{10, 20, 30, 40, 50\}$ and the regularization parameters α as $\{0.1, 0.5, 1.0\}$ and β as $\{0.0, 0.1, 0.5, 1.0\}$. For baselines, we follow earlier works and perform training over 1 epoch as it has been observed that training over multiple epochs quickly deteriorates model performance on the retain set (Yao et al., 2024a). We also conduct a parameter search for the best learning rates. All experiments are conducted on NVIDIA A100 GPUs with 40GB RAM and we report the evaluation metrics over a **single run** due to the resource-intensive nature of the experiments.

Evaluation We demonstrate the efficacy of the unlearning methods by evaluating them based on the research questions posed in Section 3.2. To quantify how well SPUL addresses RQ1, we report the accuracy and weighted F1 on the forget set, D_f^{tr}, which signifies whether the learned soft prompts can de-correlate the text features and labels. As D_f^{te} is composed of text sequences semantically or lexically similar to D_f^{tr}, the prompt tokens should result in a comparable performance decline on D_f^{te}. To evaluate SPUL on RQ2, we report performance on D_r^{tr} and consequently D_r^{te}. We require the differences in the accuracy and F1 scores of the base model before and after unlearning to be minimal for utility preservation. On the whole, we consider the tradeoff between forget and retain metrics to evaluate unlearning. To answer RQ3, we compare the number of training parameters and required GPU hours. We conduct further experiments to evaluate the influence of different loss objectives and choice of hyperparameters.

4.2 Experimental Results

Main Results We include our main results with LLaMA-2-7B in Table 2. We report performance for the original pre-trained LLM as Vanilla and the fine-tuned base model as QLoRA. We notice that the Vanilla results are considerably poorer for SST-2 than Yelp. We attribute the difference in utility to the longer and descriptive text sequences in Yelp that can provide more contextual information. Nonetheless, after fine-tuning with QLoRA, the LLM's performance increases to similar margins for both sentiment classification datasets. QLoRA fine-tuning similarly improves LLM's predictions for the WMDP+SciQ dataset. This indicates that the LLM has memorized relevant contextual information about the task and training data.

From Table 2, we observe for SST-2 that SPUL significantly reduces accuracy and F1 on D_f^{tr} compared to QLoRA demonstrating forgetting efficiency. Also, the difference in utility between SPUL and QLoRA for D_r^{tr} is minimal showing that SPUL can promote unlearning while preserving inference utility. Moreover, the metrics for D_f^{te} and D_r^{te} reflect those reported for D_f^{tr} and D_r^{tr} showing that the soft prompts effectively impose unlearning constraints on unseen samples. We observe similar performance trends for Yelp. Although the performance drop for D_f^{tr} and D_f^{te} in Yelp are not equally as large as SST-2, the forget utility with the learned tokens is significantly lesser than that of the base model. We conjecture that the additional context provided by descriptive Yelp reviews restricts the forgetting capacity of the LLM. Moreover, the utility loss in retain sets is smaller than forget sets. We also note that SPUL performs exceedingly well for the MCQA task on the WMDP+SciQ with the highest differences observed between the retain and forget metrics. Therefore, SPUL can effectively unlearn unwanted or harmful training examples in the sentiment classification and MCQA tasks.

Comparison with Baselines SPUL outperforms baselines by a significant margin; compared to GA and RL, which optimize model parameters based only on D_f^{tr}, SPUL consistently preserves utility on the retain sets with comparable or even lower

Table 2: SPUL Unlearning performance compared to baselines

Dataset	Method	Train Retain (D_r^{tr})		Train Forget (D_f^{tr})		Test Retain (D_r^{te})		Test Forget (D_f^{te})	
		ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓	ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓
SST-2	Vanilla	37.50	44.66	31.79	38.34	37.51	44.67	29.67	36.85
	QLoRA	99.89	99.89	99.72	99.72	95.57	95.57	96.07	96.07
	GA	55.66	39.80	53.93	37.83	55.96	40.16	56.89	41.25
	RL	33.31	48.08	13.82	22.97	31.00	45.56	14.26	24.18
	GA+KL	55.64	39.87	53.96	38.07	55.94	40.24	56.89	41.47
	GA+GD	97.17	97.50	13.75	20.58	94.43	94.76	11.31	17.18
	SPUL	99.15	99.39	12.98	22.94	94.93	95.24	16.07	27.42
Yelp	Vanilla	89.55	89.88	89.29	89.62	90.03	90.33	86.89	87.37
	QLoRA	99.31	99.31	99.49	99.49	98.42	98.41	98.76	98.76
	GA	66.11	63.48	67.90	64.62	65.13	62.37	67.91	64.24
	RL	53.00	67.75	52.84	66.78	52.75	67.40	49.94	65.01
	GA+KL	46.85	32.90	50.32	35.57	46.27	32.26	51.19	35.97
	GA+GD	99.23	99.42	79.69	86.98	97.76	98.00	80.90	88.19
	SPUL	89.74	93.43	55.03	70.48	89.63	93.29	60.23	74.69
WMDP+SciQ	Vanilla	47.75	46.85	26.78	17.46	46.20	46.11	23.59	14.86
	QLoRA	99.74	99.74	98.11	98.11	91.80	91.80	62.73	62.83
	GA	99.35	99.35	86.89	87.44	90.70	90.71	57.64	58.66
	RL	99.32	99.32	84.11	89.57	90.40	90.40	53.35	59.54
	GA+KL	98.84	98.85	67.44	68.91	90.20	90.24	49.33	50.26
	GA+GD	99.42	99.42	27.22	13.38	90.00	90.02	22.25	8.84
	SPUL	99.38	99.45	5.44	10.20	89.70	89.75	3.22	6.07

metrics on the forget set. For WMDP+SciQ, both GA and RL underperform in the forgetting task. GA+KL and GA+GD optimize model parameters based on both D_f^{tr} and D_r^{tr}, however, GA+KL performs poorly on all three datasets. GA+GD performs especially well on SST-2 and WMDP+SciQ but fails to enhance forget quality on Yelp which has more descriptive reviews than SST-2. SPUL can however attain effective unlearning with the least loss in utility. Across the two tasks, SPUL achieves significantly better overall trade-offs between the contrasting unlearning objectives of performance degradation and utility preservation.

Visualization We also visualize model outputs to show the effectiveness of SPUL. We utilize outputs from the last embedding layer of the LLM and map them onto a t-SNE diagram as shown in Fig. 1. The plots represent 500 data points randomly sampled from the training dataset in SST-2 for each label. In the plots, we use colors to differentiate the retain and forget examples and shapes to differentiate the positive and negative examples. We visualize the embeddings from QLoRA, i.e., the base model before unlearning and we observe a clear divide between the positively and negatively labeled samples in the embedding space. The retain and forget samples are clustered within the regions defined by each label. For the t-SNE plot of SPUL, i.e., the embeddings obtained after prepending the learned

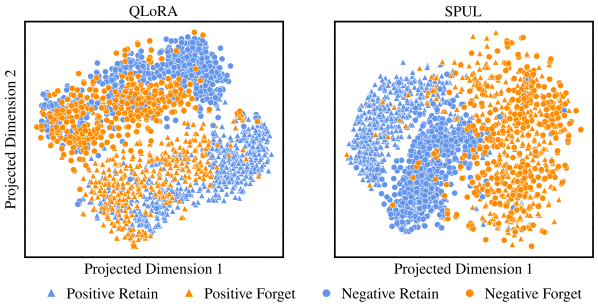

Figure 1: Embedding visualization results on SST-2 with QLoRA and SPUL

soft prompts, we notice a clear separation between the retain and forget samples as indicated by the blue and orange regions in Fig. 1. This shows that the soft prompts truly capture the differences between the forget and retain sets. Moreover, the retain samples are further grouped into clusters per their labels whereas the forget samples are mixed. This shows that the soft prompt tokens learned by SPUL successfully guide the LLM to unlearn text and label correlation for the forget samples while preserving predictive utility on the retain set.

Referring back to Table 2, SPUL metrics on D_f^{tr} and D_f^{te} closely resemble each other. We make mostly similar observations for D_r^{tr} and D_r^{te}. Our visualization results also show that the embeddings for forget samples are not distinguishable between labels. Compared to QLoRA visualization, outputs for positive and negative retain samples are closer in the embedding space. Consequently, in a black-

Table 3: SPUL performance on SST-2 across varying α and β values at $p = 30$

α	β	Train Retain (D_r^{tr})		Train Forget (D_f^{tr})		Test Retain (D_r^{te})		Test Forget (D_f^{te})	
		ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓	ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓
0.1	0.0	90.84	92.69	9.12	16.55	89.50	91.15	10.33	18.40
	0.1	92.59	93.75	6.81	12.62	90.77	91.85	10.16	18.29
	0.5	96.77	97.91	8.70	15.98	93.01	94.10	11.15	19.81
	1.0	85.19	88.00	8.49	15.47	84.64	87.19	10.66	19.02
0.5	0.0	98.17	98.69	11.86	21.17	94.34	94.87	14.59	25.07
	0.1	97.57	97.95	11.09	19.88	94.22	94.58	11.97	21.08
	0.5	97.74	98.35	13.82	24.21	93.97	94.57	17.21	29.08
	1.0	93.87	94.66	11.51	20.39	91.62	92.36	14.59	25.03
1.0	0.0	97.52	97.91	12.14	21.60	94.22	94.65	15.57	26.50
	0.1	98.64	98.96	12.14	21.54	94.63	94.97	16.07	27.41
	0.5	99.15	99.39	12.98	22.94	94.93	95.24	16.07	27.42
	1.0	95.70	96.19	14.88	25.75	93.05	93.55	17.38	29.18

Table 4: SPUL performance on SST-2 across varying sizes of forget sets

τ	Train Retain (D_r^{tr})		Train Forget (D_f^{tr})		Test Retain (D_r^{te})		Test Forget (D_f^{te})	
	ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓	ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓
25%	99.37	99.60	26.69	42.07	95.10	95.38	39.84	56.22
50%	97.66	98.47	18.96	31.78	93.80	94.62	23.61	37.60
100%	95.70	96.19	14.88	25.75	93.05	93.55	17.38	29.18

box Membership Inference Attack (MIA) (Shokri et al., 2017) scenario, it would be challenging to infer whether a particular forget sample was observed during training based only on model outputs.

Ablation Study We investigate the influence of $\mathcal{L}_r$ and $\mathcal{L}_{kl}$ and report the results in Table 3 for the SST-2 dataset. The hyperparameters α and β control the influence of the D_r^{tr} on the learned soft prompts via losses $\mathcal{L}_r$ and $\mathcal{L}_{kl}$ respectively. We fix the number of prompt tokens p at 30 and vary α in {0.1, 0.5, 1.0} and β among {0.0, 0.1, 0.5, 1.0}. From Table 3, we observe that at a fixed α, unlearning efficacy is fairly unaffected by the change in the value of β. Model utility on the retain set, however, slightly increases as β increases from 0.0 to 0.5 as $\mathcal{L}_{kl}$ gets more significance in the overall loss. Generally, the best retain performance is at β = 0.5. α, however, influences both forget and retain performance; higher α values benefit retain performance by prioritizing utility preservation whereas lower α values improve unlearning efficacy.

Hyperparameter Study We study the effect of p, the number of prompt tokens, on the effectiveness of SPUL. We fix both α and β at 1 and report results for p ranging from 10 to 50 on SST-2 in Fig. 2. We find that inference utility on retain sets D_r^{tr} and D_r^{te} is largely unaffected by the choice of p. However, we observe the most competitive forget performance at p = 30 with increasing accuracy and

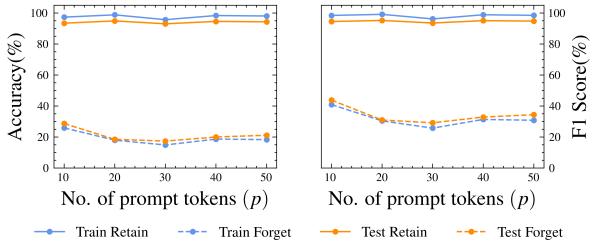

Figure 2: SPUL performance on SST-2 across varying p at $\alpha = 1$ and $\beta = 1$

F1 as p increases/decreases. We speculate that the soft prompts mostly encode information from the forget set, for instance, the forget entities in SST-2, and ultimately instruct the LLM to misclassify examples with similar encodings. Accordingly, a larger p generally benefits SPUL as shown by the decline in forget metrics but may require longer training for optimal performance.

Forget Set Size To demonstrate SPUL's stability w.r.t. the size of forget data, we evaluate it on varying sizes of the train forget set D_f^{tr} by sub-sampling τ = {25%, 50%, 100%} of the original forget set constructed for SST-2. For the remaining splits D_r^{ts}, D_f^{te}, and D_r^{te}, we use the same sets from Section 4.1 for all three configurations of D_f^{tr} to facilitate comparison. The results presented in Table 4 indicate that SPUL can achieve utility preservation across differing numbers of forget samples with minimal loss as more forget samples are added to D_f^{tr}. In contrast to the retain metrics, SPUL

Table 5: SPUL performance on SST-2 dataset using OPT-1.3B and LLaMA-2-13B

LLM	Method	Train Retain (D_r^{tr})		Train Forget (D_f^{tr})		Test Retain (D_r^{te})		Test Forget (D_f^{te})	
		ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓	ACC(%)↑	F1(%)↑	ACC(%)↓	F1(%)↓
OPT-1.3B	Vanilla	3.05	5.68	1.68	3.20	3.24	6.03	3.28	6.08
	QLoRA	99.47	99.47	99.16	99.16	95.39	95.39	95.25	95.25
	GA	79.50	78.01	70.67	67.02	78.70	77.05	71.97	67.99
	RL	55.66	39.80	53.96	37.83	55.96	40.16	56.89	41.25
	GA+KL	81.30	80.15	60.49	50.94	79.08	77.60	64.75	56.74
	GA+GD	87.56	87.59	50.53	49.07	86.47	86.50	55.25	54.59
	SPUL	94.87	96.89	16.84	28.74	91.65	93.51	17.87	29.84
LLaMA-2-13B	Vanilla	61.04	70.96	59.65	69.51	60.32	70.38	59.18	68.79
	QLoRA	99.48	99.48	99.30	99.30	96.02	96.02	95.90	95.90
	SPUL	98.87	98.93	5.97	11.25	95.50	95.60	7.38	13.54

performs better for the forget metrics when more forget samples are present in the data for SST-2. Experimental results on Yelp presented in Table 2 also highlight the robustness of SPUL against large forget sets as we assign more than 5000 samples to D_f^{tr}. As the training data contains fewer forget samples than retain samples, having a larger D_f^{tr} allows the framework to emphasize the forgetting objective thus improving the unlearning efficacy.

Results on LLaMA-2-13B and OPT-1.3B We also evaluate SPUL on OPT-1.3B and LLaMA-2-13B, with respectively fewer and almost double the parameters than LLaMA-2-7B. We fix both α and β at 1 and p at 30 and report the results for SST-2 in Table 5. We first observe that the Vanilla inference with OPT-1.3B model performs noticeably poorer than LLaMA-2-7B whereas LLaMA-2-13B improves over LLaMA-2-7B. This gap is attributed to the respective LLM's complexity which affects its generalization ability. For both OPT-1.3B and LLaMA-2-13B, SPUL can effectively achieve the forget and retain unlearning objectives as shown by the low forget accuracy and F1 compared to the retain metrics that closely resemble the base model's performance. The results also indicate larger LLMs better adapt to the unlearning task in the SPUL framework. With OPT-1.3B, SPUL notably outperforms all baseline methods. We could not run experiments on baselines with LLaMA-2-13B due to limited GPU as they require full fine-tuning. This further highlights the advantage of SPUL over baselines for parameter efficiency.

Efficiency Retraining LLMs from scratch is practically infeasible due to the associated computational cost. Fine-tuning incurs fewer resources but is expensive nonetheless. For instance, the LLM architectures used in our experiments require gradient updates for 1.42B, 6.74B, and 13B parameters for OPT-1.3B, LLaMA-2-7B, and LLaMA-2-13B respectively for fine-tuning. When $p = 30$, our SPUL reduces the computation cost by only optimizing 604K, 1.19M, and 1.49M parameters while freezing LLM parameters. Further increasing p only linearly scales the number of training parameters. We also look at the running time of SPUL on the SST-2 compared against baseline methods and find the execution time required by each model of SPUL, GA + KL, and GA+GD for one training epoch is fairly similar, around 1020 GPU seconds, as SPUL also accesses LLM parameters during backpropagation. GA and RL methods are much quicker with approximately 40 GPU seconds per epoch training time as these methods only consider the forget set. Nonetheless, SPUL avoids the overhead associated with updating LLM parameters, making it more resource-efficient.

5 Conclusion

We investigate unlearning in LLMs to remove the influence of unwanted/harmful training examples during text classification and multiple-choice question answering. We present a soft prompting strategy to unlearn subsets of training data while keeping pre-trained LLM parameters frozen to maintain the model's generalizability. The proposed SPUL framework optimizes a small number of prompt tokens using a multi-objective loss function defined on disjoint training data subsets representing the forget data subjected to removal and the retain data that aims to preserve model utility. Experimental evaluation on sentiment classification and MCQA datasets demonstrates the superior efficiency of SPUL over fine-tuning-based baselines. We also empirically show that SPUL can adapt to multiple LLMs and is robust to large unlearning requests.

Limitations

We address the limitations of this work in the following. Our experiments primarily focus on open-source LLMs as the soft prompting framework requires access to frozen pre-trained parameters to compute gradients for the soft prompts. Although, the framework avoids the overhead of updating LLM parameters. Furthermore, this work focuses on text classification and question-answering datasets for formulating and evaluating the unlearning framework. Future research could explore the efficiency of soft prompting to achieve unlearning in the context of other NLP tasks such as text generation and summarization. We further note the lack of an extensive evaluation pipeline for LLM unlearning in the current literature. Further research is needed to evaluate the robustness of LLM unlearning frameworks subject to model-stealing attacks, MIAs, and jailbreaking attempts.

Broader Impacts

In this study, our focus is to achieve LLM unlearning in a resource-efficient manner. We aim to enable forgetting of unwanted or harmful knowledge in a pre-trained LLM while maintaining model efficiency to avoid exploitation of sensitive information. The datasets used for evaluation are publicly available and implemented within their intended use. Our usage of publicly available pre-trained LLMs also adheres to the associated licenses. We hope our study can further the research and literature on resource-efficient LLM unlearning.

References

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *CoRR*.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*.

Micheal Grynbaum and Ryan Mac. 2023. The times sues openai and microsoft over a.i. use of copyrighted work. The New York Times.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. 2020. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research.

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 (OPTvolume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. 2024. Soul: Unlocking the power of second-order optimization for llm unlearning. *Preprint*, arXiv:2404.18239.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Kiran Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International*

Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (OPTvolume 1: Long Papers), Virtual Event, August 1-6, 2021*.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (OPTvolume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. TOFU: A task of fictitious unlearning for llms. *CoRR*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *CoRR*.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2024. In-context unlearning: Language models as few-shot unlearners. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. 2023. Knowledge unlearning for llms: Tasks, methods, and challenges. *CoRR*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*.

Pratiksha Thaker, Yash Maurya, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. *CoRR*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *CoRR*.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024b. Large language model unlearning. *Preprint*, arXiv:2310.10683.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*.