Soft Prompting for Unlearning in Large Language Models

Anonymous ACL submission

Abstract

The widespread popularity of Large Language Models (LLMs), partly due to their unique ability to perform in-context learning, has also brought to light the importance of ethical and safety considerations when deploying these pre-trained models. In this work, we focus on investigating machine unlearning for LLMs motivated by data protection regulations. 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 the unlearning of a subset of training data. With losses designed to enforce forgetting as well as utility preservation, our framework **S**oft **P**rompting for **Unl**earning (SPUL) learns prompt tokens that can be appended to an arbitrary query to induce unlearning of specific examples at inference time without updating LLM parameters. We conduct a rigorous evaluation of the proposed method and our results indicate that SPUL can significantly improve the trade-off between utility and forgetting in the context of text classification with LLMs. We further validate our method using multiple LLMs to highlight the scalability of our framework and provide detailed insights into the choice of hyperparameters and the influence of the size of unlearning data. Code and data are available at `https://tinyurl.com/softprompt`.

1 Introduction

With advancements in transformer models (Vaswani et al., 2017) and the availability of massive text corpus, language models have rapidly evolved over the past decade. 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, ethical and security concerns have been raised due to the inclusion of private and sensitive information in the training data. For example, LLMs can regurgitate individual 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 the previously outlined issues resulting from the presence of sensitive 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 this work, we propose a novel approach to unlearning in LLMs via soft prompting which is less resource-intensive than fine-tuning. To the best

of our knowledge, this is the first work to investigate the use of soft prompting for unlearning in LLMs. Soft prompting simplifies the process of adapting LLMs to an arbitrary downstream task by optimizing learnable token embeddings that encode signals from a corresponding dataset (Lester et al., 2021; Li and Liang, 2021). The soft prompts are trained end-to-end and essentially act as instructions for a frozen pre-trained LLM during inference. We leverage this ability to modulate LLM outputs using prompts and formulate **S**oft **P**rompting for **Unl**earning (SPUL), a resource-efficient mechanism to achieve LLM unlearning in text classification. We optimize a set of soft prompt parameters that 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 the learning of soft prompts. 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, our framework can utilize the same pre-trained LLM for different unlearning tasks and datasets during inference.

We evaluate SPUL for sentiment classification on benchmark NLP datasets when unlearning a subset from the corresponding training dataset and compare against various fine-tuning-based methods. We show that SPUL can effectively induce forgetting during inference while preserving the pre-trained utility with significant improvements over baselines. 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 of different parameter sizes and different 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. The idea was conceived by Lester et al. (2021); they added trainable continuous embeddings to the encoder input sequence of an LLM and showed that the learned prompts achieve 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 showed that their method is comparable to fine-tuning approaches for text generation tasks. Liu et al. (2021) concatenated trainable continuous prompts with discrete prompts along with a prompt encoder module 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., 2022a).

2.2 Unlearning in LLMs

Machine unlearning arose as a promising solution to address 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; Liu et al., 2022b; Guo et al., 2020; Sekhari et al., 2021; Golatkar et al., 2020). In the context of LLMs, machine unlearning is quickly gaining prominence due to concerns stemming from 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 for queries related to Harry Potter with outputs containing generic translations. In contrast to these works, Jia et al. (2024) utilize similar fine-tuning objectives but 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 rep-

resent information about 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 data that can be unlearned. For instance, Pawelczyk et al. (2023) perform in-context unlearning by crafting input comprised of unlearn samples paired with flipped labels and other demonstrations with correct 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 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 classification 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 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 and re-training?

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} for text classification. 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 misclassification 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* or *unknown* 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 our SPUL framework, the prepended prompt tokens $\boldsymbol{\phi}$ 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

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

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. We further constrain the predictive distribution of the base model such that $h_\theta(\{\boldsymbol{\phi}, \mathbf{x}_j\})$ reflects $h_\theta(\mathbf{x}_j)$ for any $\mathbf{x}_j \in D_r^{tr}$. We quantify this difference using KL divergence as

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

where $\mathrm{KL}(\cdot)$ denotes the KL divergence term. $h_\theta(\{\boldsymbol{\phi}, \mathbf{x}_j\})$ 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 $\boldsymbol{\phi}$ 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 two standard NLP datasets SST-2 (Socher et al., 2013) and Yelp polarity (Zhang et al., 2015) for the task of sentiment classification. The datasets 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 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 first 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}. After preprocessing, the constructed sets $D_f^{tr}/D_r^{tr}/D_f^{te}/D_r^{te}$ contains $1425/46331/610/19855$ samples. For the Yelp polarity dataset, 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. For Yelp, $D_f^{tr}/D_r^{tr}/D_f^{te}/D_r^{te}$ includes $5081/95012/885/18089$ samples.

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 on the forget set by maximizing the cross-entropy loss in place of the standard minimization. Fine-tuning with Random Labels (RL) (Golatkar et al., 2020; Yao et al., 2024a) similarly optimizes the base model on the forget set but by enforcing convergence on random labels. 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) integrate parameter optimization using the retain set with GA to balance forgetting effectiveness with utility (Yao et al., 2024a). The former defines a KL-divergence constraint on the LLM's output distribution and the latter implements the standard cross-entropy loss. Note that for all four baselines, we perform full fine-tuning of the LLM 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 our SPUL frame-

Table 1: 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

work. We further validate the unlearning effectiveness of our method 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 (2) epochs on SST-2 (Yelp) 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. As for the configurations of SPUL, we fix the learning rate at 0.0001 across all LLMs, datasets and vary prompt token length p among {10, 20, 30, 40, 50}. We also vary the regularization parameters α as {0.1, 0.5, 1.0} and β as {0.0, 0.1, 0.5, 1.0}[1]. We train our unlearning framework for a total of 10 epochs. As for baseline model specifications, we conduct a parameter search for the best learning rates and report the most competitive results after 1 epoch of training. 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 framework by evaluating the methods based on the research questions posed in Section 3.2. To quantify how well our SPUL framework 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 based on RQ2, we report model performance on D_r^{tr} and consequently D_r^{te}. We emphasize the differences in the accuracy and F1 scores of the base model before and after unlearning to signify utility preservation. Finally, to answer RQ3, we report the number of training parameters and required GPU hours and compare them against baseline metrics.

[1]We note that advanced approaches, e.g., utility function, Pareto-based, and constraint-based methods, can be potentially adopted to determine values of α and β.

4.2 Experimental Results

Main Results We include our main results with LLaMA-2-7B in Table 1. We report performance metrics for the original pre-trained LLM denoted as Vanilla and the fine-tuned base model denoted as QLoRA. We notice that the Vanilla results are considerably poorer for SST-2 compared to Yelp which validates our setup of fine-tuning the original LLM on the datasets for memorization. We attribute the difference in utility to the fact that the text sequences in Yelp are significantly longer and provide more contextual information. Nonetheless, after fine-tuning with QLoRA, the LLM's performance increases to similar margins for both datasets.

From Table 1, we observe that SPUL significantly reduces accuracy and F1 on D_f^{tr} compared to QLoRA demonstrating forgetting efficiency. At the same time, the difference in utility between SPUL and QLoRA for D_r^{tr} is minimal showing that our method can promote unlearning while also 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 samples unseen during training. We observe similar performance

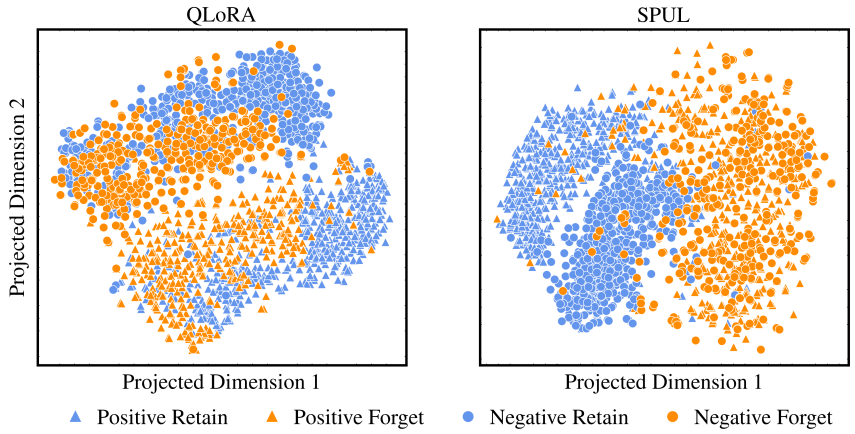

Figure 1: Embedding visualization results on SST-2 with QLoRA and SPUL

trends for Yelp. Although the performance drop for D_f^{tr} and D_r^{te} are not equally as large as SST-2, the forget utility with the learned tokens is significantly lesser in comparison to the base model. We conjecture that the additional context provided by descriptive Yelp reviews restricts the forgetting capacity of the LLM. Also point out that utility loss in retain sets is smaller than forget sets.

Furthermore, SPUL outperforms baseline methods by a significant margin; compared to GA and RL, which optimize model parameters based only on the D_f^{tr}, SPUL consistently preserves inference utility on the retain sets with comparable or even lower metrics on the forget set. GA+KL and GA+GD optimize model parameters based on both D_f^{tr} and D_r^{tr}. However, GA+KL performs poor on both datasets. GA+GD performs especially well on SST-2 but fails to enhance forget quality on Yelp which has more descriptive reviews. The proposed SPUL framework can however attain effective unlearning with the least loss of model utility. Among the compared methods, 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 our SPUL method. 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 use 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 together within the regions defined by each label. For the t-SNE plot of SPUL, i.e., the embeddings obtained after pretending the learned 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. On the other hand, the positive and negative forget samples are mixed together. 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 1, SPUL metrics on D_f^{tr} and D_f^{te} closely resemble each other for both SST-2 and Yelp. We make similar observations for D_r^{tr} and D_r^{te}. Our visualization results also show that the output embeddings for forget samples are not distinguishable between labels. Compared to QLoRA visualization, model outputs for positive and negative retain samples are closer in the embedding space as well. As a result, in a black-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.

Hyperparameter Study We conduct a series of experiments to investigate the influence of the hyperparameters α and β on the unlearning performance of the proposed SPUL framework and report the results in Table 2 for the SST-2 dataset. The hy-

Table 2: 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 3: 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

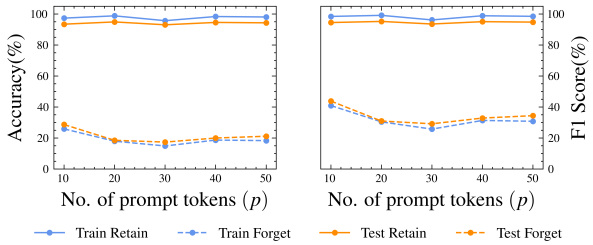

Figure 2: SPUL performance on SST-2 across varying p at $\alpha = 1$ and $\beta = 1$

perparameters control the influence of the retain set 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 for all results and vary α in {0.1, 0.5, 1.0} and β among {0.0, 0.1, 0.5, 1.0}. From Table 2, 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. We generally observe the best retain performance at $\beta = 0.5$. The value of α influences performance on both forget and retain sets; higher α values benefit retain performance by prioritizing utility preservation whereas lower α values improve unlearning efficacy.

We also study the effect of the number of prompt tokens, represented by p, on the unlearning effectiveness of SPUL. We fix both α and β at 1 and run experiments with p ranging from 10 to 50 on SST-2 and report results in Fig. 2. We find that inference utility on retain sets D_r^{tr} and D_r^{te} is largely unaffected by the different choice of p. However, we observe the most competitive forget performance at p = 30 with increasing accuracy and F1 as p increases/decreases. We speculate that the soft prompts mostly encode information from the forget set, for instance, the named entities in SST-2 whose reviews are unlearned, and ultimately instruct the LLM to misclassify examples with similar encodings. Accordingly, a larger p generally benefits our soft prompting framework as made evident by the decline in forget metrics but may require longer training for optimal performance.

Forget Set Size To demonstrate the stability of our method w.r.t. the size of forget data, we evaluate SPUL 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 test forget set D_f^{te} and the retain sets D_r^{ts} and D_r^{te}, we use the same sets defined in Section 4.1 for all three configurations of D_f^{tr} to facilitate comparison. We present the results from this experiment on SST-2 in Table 3. Our results indicate that SPUL can achieve utility preservation across differing numbers of forget samples with minimal loss as

Table 4: 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
	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

more forget samples are added to D_f^{tr}. In contrast to the retain metrics, SPUL clearly 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 1 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 comparatively 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 additionally evaluate the unlearning efficacy of our SPUL on different LLMs. In particular, we purposely choose OPT-1.3B with fewer parameters and LLaMA-2-13B with almost double the parameters compared to LLaMA-2-7B. In addition to the unlearning efficacy, this study also evaluates the scalability of our SPUL framework. We fix the hyperparameters α and β at 1 and p at 30 and report the results for SST-2 in Table 4. We first observe that the Vanilla inference with OPT-1.3B model performs noticeably poorer than LLaMA-2-7B whereas LLaMA-2-13B significantly improves over the initial metrics. This may be attributed to the pre-trained models' complexity which affects their generalization ability. We similarly perform fine-tuning using QLoRA to ensure the unlearning dataset has been memorized by the respective LLM. Moreover, SPUL can effectively achieve the forget and retain unlearning objectives as made evident by the low forget accuracy and F1 compared to the retain metrics that closely resemble the base model's performance. The results also indicate that the larger the LLM, the better it adapts to the unlearning task in our SPUL framework.

Efficiency For LLMs, retraining from scratch is practically infeasible due to computational time and resources required for a huge set of parameters. Although fine-tuning pre-trained LLMs incurs less costs than retraining, the cost is still high. 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 when implementing unlearning based on 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 approximate 40 GPU seconds of 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

In this work, we investigate unlearning in LLMs to remove the influence of unwanted training examples during text classification. 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. Our 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 that is subjected to removal and the retain data that aims to preserve model utility. Experimental evaluation on sentiment classification datasets demonstrates the superior efficiency of our soft prompting-based unlearning over fine-tuning-based baselines. We also empirically show that SPUL can adapt to multiple LLMs and is robust to a high number of unlearning samples.

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 despite not needing to update the LLM parameters. Furthermore, this work focuses on the task of text classification, specifically sentiment classification for the formulation of the unlearning framework and evaluation. Future research could explore the efficiency of soft prompting to achieve unlearning in the context of NLP tasks such as text generation, question answering, text summarization, and so on. Also, the soft prompting unlearning framework has not been evaluated comprehensively as we emphasize performance metrics to demonstrate unlearning efficacy. We note that there is a lack of an extensive evaluation pipeline for LLM unlearning in the current literature. Further research is needed to evaluate the robustness of the framework 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 and undesirable knowledge as per users' requests while maintaining model efficiency to avoid exploitation of protected information. The datasets used for evaluation are publicly available and implemented within the 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*, pages 141–159. IEEE.

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*, pages 463–480. IEEE Computer Society.

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*, pages 12041–12052. Association for Computational Linguistics.

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*, abs/2310.02238.

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*, pages 9301–9309. Computer Vision Foundation / IEEE.

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*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14389–14408. Association for Computational Linguistics.

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*, pages 3045–3059. Association for Computational Linguistics.

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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 4582–4597. Association for Computational Linguistics.

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*, abs/2402.08787.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022a. 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 (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 61–68. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. 2022b. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, London, United Kingdom, May 2-5, 2022*, pages 1749–1758. IEEE.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. TOFU: A task of fictitious unlearning for llms. *CoRR*, abs/2401.06121.

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*, abs/2311.17035.

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

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2023. In-context unlearning: Language models as few shot unlearners. *Preprint*, arXiv:2310.07579.

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*, pages 18075–18086.

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*, pages 3–18. IEEE Computer Society.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. 2023. Knowledge unlearning for llms: Tasks, methods, and challenges. *CoRR*, abs/2311.15766.

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*, pages 1631–1642. ACL.

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

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*, pages 5998–6008.

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*, pages 1322–1338. Association for Computational Linguistics.

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*, pages 2875–2886. Association for Computational Linguistics.

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

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*, pages 6032–6048. Association for Computational Linguistics.

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*, pages 649–657.