

# Selective Forgetting: Advancing Machine Unlearning Techniques and Evaluation in Language Models

Lingzhi Wang<sup>1</sup>, Xingshan Zeng<sup>2\*</sup>, Jinsong Guo<sup>3</sup>, Kam-Fai Wong<sup>4,5</sup>, Georg Gottlob<sup>6</sup>

<sup>1</sup>Harbin Institute of Technology, Shenzhen, China

<sup>2</sup>Huawei Noah's Ark Lab, China <sup>3</sup>Unlimidata Ltd, United Kingdom <sup>4</sup>The Chinese University of Hong Kong, China

<sup>5</sup>MoE Key Laboratory of High Confidence Software Technologies, China <sup>6</sup>University of Calabria, Italy

wanglingzhi@hit.edu.cn, zeng.xingshan@huawei.com, jinsong.guo@unlimidata.com, kfwong@se.cuhk.edu.hk, georg.gottlob@unical.it

## Abstract

This paper explores Machine Unlearning (MU), an emerging field that is gaining increased attention due to concerns about neural models unintentionally remembering personal or sensitive information. We present SEUL, a novel method that enables selective and fine-grained unlearning for language models. Unlike previous work that employs a fully reversed training objective in unlearning, SEUL minimizes the negative impact on the capability of language models, particularly in terms of generation. Furthermore, we introduce two innovative evaluation metrics, sensitive extraction likelihood (S-EL) and sensitive memorization accuracy (S-MA), specifically designed to assess the effectiveness of forgetting sensitive information. In support of the unlearning framework, we propose efficient automatic online and offline sensitive span annotation methods. The online selection method, based on language probability scores, ensures computational efficiency, while the offline annotation involves a two-stage LLM-based process for robust verification. In summary, this paper contributes a novel selective unlearning method (SEUL), introduces specialized evaluation metrics (S-EL and S-MA) for assessing sensitive information forgetting, and proposes automatic online and offline sensitive span annotation methods to support the overall unlearning framework and evaluation process.

## Introduction

Machine Unlearning (MU) (Romero, Barrio, and Belanche 2007; Karasuyama and Takeuchi 2009; Cao and Yang 2015) has increasingly attracted the attention of researchers. The focus on MU stems primarily from the fact that neural models are trained on data mainly sourced from the Internet, and the trained model may permanently “remember” personal or sensitive information contained in the training data. Concerns about the leakage of personal sensitive data from neural networks have intensified, especially following the breakthrough of language models, which exhibit incredible capabilities in data generation. Meanwhile, the “right to be forgotten” has been legislated in many countries, such as the General Data Protection Regulation (GDPR) in the European Union and the PIPEDA privacy legislation in Canada.

\* Xingshan Zeng is the corresponding author.

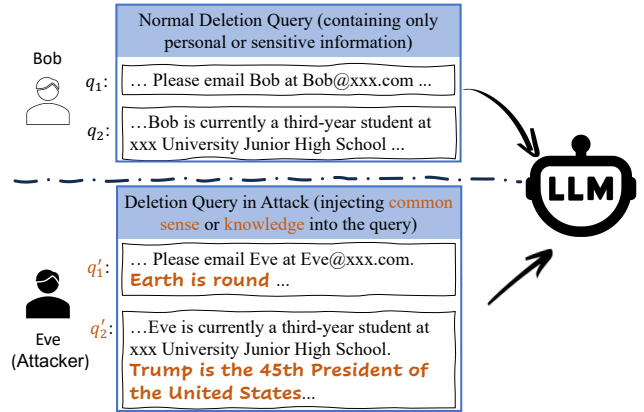


Figure 1: Illustration of knowledge injection attack.

This right mandates that companies erase personal data upon user request. Furthermore, while removing data from back-end databases is straightforward, it poses a challenge for neural models as the relationship between the model weights and the data is unclear.

Given the identified necessities and challenges in unlearning, particularly in the context of large language models, researchers have focused on machine unlearning to make trained models forget specific data. While many prior works (Golatkar, Achille, and Soatto 2020a,b; Mehta et al. 2022) in machine unlearning address computer vision classification tasks, fewer target generation tasks in NLP. Wang et al. (2023) propose a general machine unlearning framework, but it relies on additional model training, which is costly for language models. Meanwhile, Jang et al. (2023) introduces sequential knowledge unlearning for language models, utilizing a reversed language modeling learning objective. However, employing a fully reversed training objective in unlearning can significantly impact the language model's generation capability.

In contrast to Jang et al. (2023), which fully reverses the training loss of instances for forgetting, we propose a selective unlearning method, SEUL. SEUL achieves knowledge forgetting in a fine-grained manner, focusing on specific sequence spans rather than entire instances, as illustrated in Fig. 1. In the scenarios where an attacker (e.g., Eve in Fig. 1) injects general knowledge into regular personal data and re-

quests data forgetting, fully reversed unlearning can compromise the model. Our selective unlearning method, however, better preserves the generation capability of language models, especially when most parts of the sequence to be deleted consist of general expressions. Beyond extreme adversarial cases, fully reversed unlearning can also detrimentally impact the capability of language models as the number of deletion queries increases. This is due to its inherent nature of affecting model’s training on all tokens in deletion queries without selection.

Moreover, we introduce automatic online and offline methods for sensitive span annotation to facilitate the training and evaluation of SEUL. Online selection relies on language probability scores of tokens, ensuring efficiency in calculation and supporting selective unlearning without additional dependencies. For offline annotation, we employ a two-stage process with a large language model (LLM, e.g., ChatGPT), ensuring thorough annotation and verification. The offline annotated spans serve to evaluate unlearned language models, accompanied by two novel unlearning evaluation metrics—sensitive extraction likelihood (S-EL) and sensitive memorization accuracy (S-MA)—specifically designed to address sensitive information unlearning.

In brief, the main contributions of this paper are:

- We propose a novel unlearning method called SEUL, which facilitates selective, fine-grained, and effective unlearning in language models. It shows a comparable ability to maintain classification performance and a significantly enhanced capability in preserving generation performance after unlearning, as compared to the SOTA.
- We propose two new evaluation metrics, namely sensitive extraction likelihood (S-EL) and sensitive memorization accuracy (S-MA), to emphasize the assessment of unlearning methods based on their ability to forget sensitive information rather than general information.
- We propose both online and offline automatic sensitive information annotation methods to assist the effective selective unlearning framework and facilitate efficient unlearning evaluation.

## Related Work

We provide a comprehensive overview of machine unlearning, focusing on three key aspects: methodology, datasets, and evaluation metrics.

**Methodology.** Machine unlearning falls into two categories: Exact Unlearning and Approximate Unlearning. Exact Unlearning (Cao and Yang 2015; Ginart et al. 2019; Bourtole et al. 2021) aims to completely eliminate the impact of deleted data but struggles with scalability in deep neural networks (Cao and Yang 2015; Ginart et al. 2019) and efficiency (Bourtole et al. 2021). In contrast, Approximate Unlearning methods like Golatkar, Achille, and Soatto (2020a), Guo et al. (2019), Koh and Liang (2017), and Mehta et al. (2022) prioritize efficiency at the cost of exactness. Though most methods are developed in computer vision, there are some recent unlearning works in natural language (Wang et al. 2023; Jang et al. 2023). Wang et al.

(2023) proposes a general framework that relies on extra model training to maintain knowledge gap. However, training extra models is expensive and impractical for large language models. Jang et al. (2023) simply negates the training objective.

**Datasets.** Currently, no dedicated datasets exist for the explicit examination and evaluation of unlearning methods. Researchers typically assess the efficacy and efficiency of unlearning approaches across diverse datasets based on individual considerations. For instance, Lu et al. (2022) investigate toxicity unlearning using the RealToxicityPrompts (Liu et al. 2021) and WritingPrompts (Fan, Lewis, and Dauphin 2018) datasets. In another study, Wang et al. (2023) perform experiments on three representative datasets: LEDGAR (Tuggenier et al. 2020), IWSLT14 German-English (Cettolo et al. 2014), and PersonaChat (Zhang et al. 2018). This evaluation encompasses classification, translation, and generation tasks. Additionally, Jang et al. (2023) conduct experiments on reasoning datasets, such as Hellaswag (Zellers et al. 2019) and MathQA (Amini et al. 2019), to assess language models’ capabilities before and after the unlearning process.

**Evaluation Metrics.** We mainly discuss the unlearning evaluation in natural language field here. Patil, Hase, and Bansal (2023) assess unlearning performance using  $\Delta$ -Acc, comparing accuracy scores before and after data point deletion on CounterFact (Meng et al. 2022) and zsRE (Levy et al. 2017). Jang et al. (2023) introduce Extraction Likelihood (EL) and leverage Memorization Accuracy (MA) from (Tirumala et al. 2022) to quantitatively measure information leakage during extraction attacks. Additionally, Wang et al. (2023) evaluate unlearning through Jensen–Shannon Divergence (JSD), Language Model Probability Distance (LPD), and Proportion of instances with Decreased Language Model Probability (PDL). It is noteworthy that this method heavily relies on retraining models to assess changes, which can be prohibitively costly, especially in the case of large language models.

## Selective Unlearning For LMs

### Our Unlearning Methodology

**Problem Formulation** By following Wang et al. (2023), we formulate machine unlearning as follows. Given a training set  $D \in \mathcal{Z}^*$ , The process of training a model on data set  $D$  is denoted by a function  $A : \mathcal{Z}^* \rightarrow \mathcal{H}$ , where  $\mathcal{Z}$  is the space of data instances and  $\mathcal{H}$  is the hypothesis space of models. Then the trained model can be denoted as  $A(D)$ . For general machine unlearning, the unlearning mechanism is denoted as a function  $U$ , which takes a training dataset  $D \in \mathcal{Z}^*$  (optional), a forget set  $D_f \subset D$  (containing data to be unlearned) and a model  $A(D)$  as input, and returns an unlearned model  $U(D, D_f, A(D)) \in \mathcal{H}$ .

For our selective unlearning, for each instance in the forget set  $D_f$ , we define a forget span set that explicitly refers to the sequence spans that need to be unlearned. Formally, for any  $x = (x_1, x_2, \dots, x_T) \in D_f$ , the forget span set is denoted as  $s^x = (s_1, s_2, \dots, s_m)$ , where  $m$  is the number of

the forget spans and  $s_i$  represents a continuous sub-sequence of  $x$ , i.e.,  $s_i = (x_{j_i}, x_{j_i+1}, \dots, x_{j_i+|s_i|-1})$ , where  $j_i$  indicates the original index in  $x$  of the first token of  $s_i$ .

**Our SEUL Unlearning Method** Different from Jang et al. (2023) which simply negates the original training objective of minimizing the negative log-likelihood of token sequences as an unlearning method, we propose a selective unlearning method called SEUL. SEUL forgets knowledge in a fine-grained manner, aiming to forget specific sequence spans instead of instance-level forgetting. The motivation mainly comes from the fact that instance-level forgetting cannot handle the adversarial attack mentioned in Fig. 1 and selective unlearning can have less impact on the performance of the original language model. The learning objective of SEUL is to *minimize* the following loss function for all  $x \in D_f$ :

$$\mathcal{L}_{UL}(A(D), x) = \sum_{s_i \in s^x} \sum_{t=j_i}^{j_i+|s_i|-1} \log(p_\theta(x_t|x_{<t})) \quad (1)$$

where  $x_{<t}$  denotes the sequence preceding the index  $t$ , and  $j_i$  is the start index of subsequence  $s_i$  in original sequence  $x$ .  $p_\theta(x_t|x_{<t})$  denotes the conditional probability of predicting the next token when given prefix  $x_t$  to an LM  $A(D)$  with parameters  $\theta$ . With the learning objective, the model is pushed to only unlearn the forget spans in forget span set  $s^x$  without affecting other general knowledge.

### How to Determine the Forgetting Span?

It’s essential to note that, the forgetting span could be tailored based on practical needs (e.g., for spans with toxic words) in some scenarios. However, a more common scenario is that the forget span set is not predefined, and we must develop a span selection/annotation mechanism to achieve the goal of selective unlearning. We propose a comprehensive span selection mechanism designed to adapt to various stages of machine unlearning, specifically addressing concerns related to privacy leakage. In the following sections, we provide details about our forget span selection/annotation design.

**Online Selection** When conducting unlearning based on our proposed learning objective (i.e., Eq. 1) without providing forget spans but for general goal of forgetting (i.e., reducing the privacy leakage risk of  $D_f$ ), we propose an automatic and efficient target span online selection method. We assume that content spans with privacy leakage risks are most likely not common knowledge (i.e. the trained model should not have seen them frequently during training). This means that they are probably with low language probability in language models (a higher perplexity score accordingly). Based on the assumption, we thereby define the following online selection process:

$$\text{Select}(x, \alpha) = \{x_t \mid \log(p'_\theta(x_t|x_{<t})) < \alpha\} \quad (2)$$

where  $p'_\theta(\cdot)$  indicates the language probability of original model (before unlearning) and  $\alpha$  should be a predefined language probability threshold or the average log-probability of

the tokens in  $x$  (i.e.  $\alpha = \sum_{x_t \in x} \log(p'_\theta(x_t|x_{<t}))$ ). We also include tokens that are located between two closely positioned tokens (for any  $x_i$  and  $x_j$ , we consider them close if  $|i-j| \leq 2$ ). This decision is based on our preliminary observation that, in some cases, a complete word or phrase may exhibit a higher language probability in the middle tokens. Then we aggregate all the selectively adjacent tokens into spans, forming  $s^x = (s_1, s_2, \dots, s_m)$ . The efficient selection of forget spans described above can be conducted during the training of the language model, thereby facilitating our proposed selective unlearning. As this selection is based on our aforementioned assumption, we will empirically discuss its rationality in the Experimental Results Section.

**Offline Annotation** Online selection provides a rapid and cost-effective solution, but it may lack the precision required for identifying genuinely privacy-sensitive tokens. To address this limitation, we propose an offline forget spans annotation method that ensures greater accuracy. The annotation workflow and the original forget set  $D_f$  together with the annotated forget span sets could serve as a valuable resource to evaluate unlearning methods’ efficiency in forgetting target knowledge. In the following, we describe our bi-directional verified LLM-based annotation process.

(i) Forward Span Annotation. We leverage the strong in-context learning capability of large language models to facilitate forget span annotation. As previously mentioned, this work primarily focuses on the general purpose of unlearning, i.e., forgetting information with high risks of leaking private information. To achieve this, we instruct large language models (e.g., ChatGPT) to annotate sensitive spans in a given text using a few-shot setting. The detailed prompt is provided in the Appendix. Following this prompt, the large language models generate sensitive spans<sup>1</sup> for each instance  $x$  in the forget set  $D_f$ . The entire process can be summarized as follows:

$$\check{s}^x \leftarrow \mathcal{F}(x, \mathcal{D}) \quad (3)$$

where  $\mathcal{D}$  denotes the examples used in few-show learning.

(ii) Backward Verification. Previous research (Ling et al. 2023; Shinn et al. 2023) has demonstrated double-checking the reasoning results generated by LLMs can produce significantly more reliable content. In this work, we employ a backward verification mechanism. Instead of merely prompting LLMs to validate the accuracy of previously generated sensitive spans, we instruct them to independently assess and score the generated spans on a scale of  $\{0, 1, 2\}$  without providing the entire sequence (i.e.,  $x$ ). Subsequently, we filter out spans with low scores (i.e., score 0).

### Evaluation of Unlearning

In this subsection, we present two newly proposed quantitative evaluation metrics designed for assessing machine unlearning with a specific focus on handling sensitive information: Sensitive Extraction Likelihood (S-EL) and Sensitive Memorization Accuracy (S-MA).

<sup>1</sup>An exact token matching-based post-processing is applied to ensure that the generated spans match the original text input.

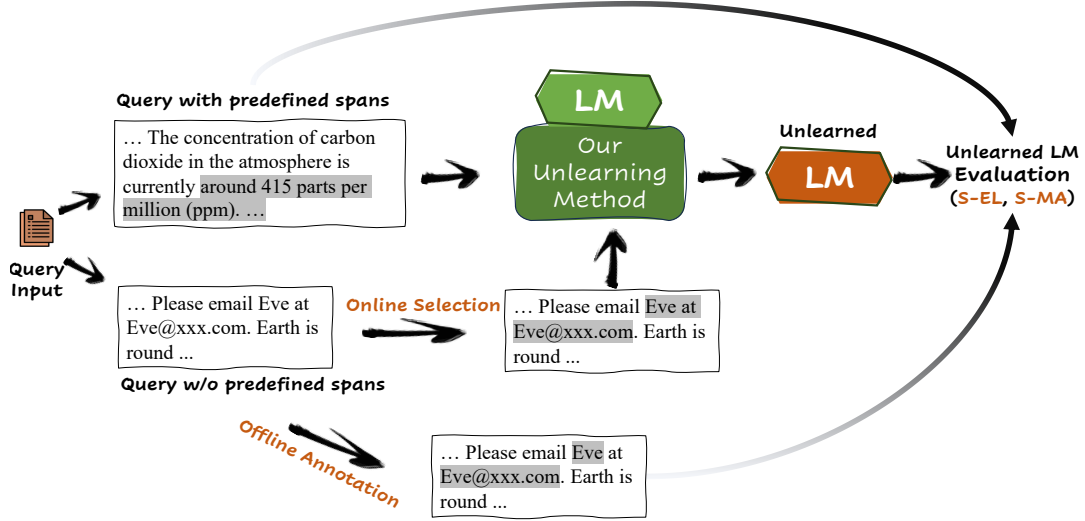


Figure 2: Workflow of SEUL: Queries with predefined spans (either sensitive or within other definitions) can be inputted directly into SEUL. For queries without predefined spans, we conduct online selection before feeding them to SEUL.

**Sensitive Extraction Likelihood** Before introducing S-EL, we provide a succinct definition of string overlap (OVL) (following Jang et al. (2023)) and our proposed overlap with consideration for sensitive information (S-OVL) given two token sequences  $a$  and  $b$ :

$$\text{OVL}_n(a, b) = \frac{\sum_{c \in n\text{-grams}(a)} \mathbb{1}\{c \subseteq b\}}{|n\text{-grams}(a)|} \quad (4)$$

$$\text{S-OVL}_n(a, b) = \frac{\sum_{c \in n\text{-grams}(a)} \mathbb{1}\{(c \cap s^b) \neq \emptyset \wedge (c \cap s^b) \subseteq b\}}{\sum_{c \in n\text{-grams}(a)} \mathbb{1}\{(c \cap s^b) \neq \emptyset\}} \quad (5)$$

where  $\mathbb{1}\{\cdot\}$  denotes an indicator function that returns 1 if the evaluated condition is true; otherwise 0.  $n\text{-grams}()$  returns the list of  $n$ -grams in the given token sequence, and  $s^b$  represents the sensitive span set in  $b$  that needs to be unlearned.  $c \subseteq b$  indicates that  $c$  is the subsequence of  $b$ .  $c \cap s^b$  returns the common subsequence of  $c$  and any span in  $s^b$ . Notably, the distinction between S-OVL and OVL lies in the fact that S-OVL exclusively considers  $n$ -grams containing sensitive spans.

Building upon S-OVL, we propose S-EL, which assesses the likelihood of a language model  $f_\theta$  generating sensitive information when prompted with a prefix in  $x$ , in the basis of  $n$ -gram.

$$\text{S-EL}_n(x) = \frac{\sum_{t=1}^{T-n} \text{S-OVL}_n(f_\theta(x_{<t}), x_{\geq t})}{T-n} \quad (6)$$

where  $f_\theta(x_{<t})$  denotes the suffix generated by the language model  $f_\theta$  given the prefix  $x_{<t}$ .

**Sensitive Memorization Accuracy** S-MA quantifies the model’s accuracy in predicting the next token given a prefix following Tirumala et al. (2022), but only considering the

sensitive information. We define it as follows.

$$\text{S-MA}(x) = \frac{\sum_{t=1}^{T-1} \mathbb{1}\{\text{argmax}(p_\theta(\cdot|x_{<t})) = x_t \wedge x_t \in s^x\}}{\sum_{t=1}^{T-1} \mathbb{1}\{x_t \in s^x\}} \quad (7)$$

where  $\text{argmax}(p_\theta(\cdot|x_{<t}))$  denotes the most probable next token predicted by the language model.  $T$  is the total number of tokens in the sequence  $x$  and  $x_t \in s^x$  indicates that token  $x_t$  is contained in any sensitive span of span set  $s^x$ .

Both S-EL and S-MA provide a robust framework for evaluating the efficacy of machine unlearning, particularly in contexts where the handling of sensitive information is a critical concern. For comparison, we also denote the extraction likelihood and memorization accuracy without considering whether the tokens are sensitive or not as  $\text{EL}_n$  (Jang et al. 2023) and MA (Tirumala et al. 2022) and display their results together with the two proposed metrics.

## Experimental Setup

### Datasets and Evaluation Metrics

As we focus on unlearning of language models, we start from pretrained language models, then we do unlearning on forget dataset  $D_f$  and test the models (original LMs and unlearned LMs) on test set  $D_t$ .

**Forget Dataset  $D_f$ .** In order to ensure a fair comparison with Jang et al. (2023) and to assess the privacy risk associated with language models, we employ the identical samples as those disclosed by Jang et al. (2023). The forget set is sourced from the Training Data Extraction Challenge<sup>2</sup>, comprising 15,000 examples, each consisting of 200-token sequences from various domains of Pile corpora.

**Evaluation Datasets  $D_t$ .** To evaluate general language modeling capabilities, we employ 8 classification tasks (i.e.,

<sup>2</sup><https://github.com/google-research/lm-extraction-benchmark>

| Models          |          | Forget Evaluation |      |                        |             | Avg. 8 Cla. | Avg. 4 Dia. |              | Epoch |
|-----------------|----------|-------------------|------|------------------------|-------------|-------------|-------------|--------------|-------|
|                 |          | EL <sub>10</sub>  | MA   | S-EL <sub>10</sub> (↓) | S-MA(↓)     | Acc(↑)      | F1(↑)       | PPL(↓)       | #     |
|                 | Original | 31.5              | 76.9 | 22.0                   | 62.4        | 43.7        | 7.4         | 43.6         | -     |
| GPT-Neo<br>125M | KUL      | 0.2               | 26.8 | 1.3                    | 22.9        | 40.9        | 1.1         | 539.1        | 15.6  |
|                 | SEUL     | 1.0               | 29.5 | <b>0.3</b>             | <b>16.1</b> | <b>41.0</b> | <b>7.8</b>  | <b>179.8</b> | 16.2  |
|                 | Original | 60.4              | 91.1 | 44.6                   | 81.6        | 48.6        | 12.3        | 26.0         | -     |
| GPT-Neo<br>1.3B | KUL      | 0.3               | 27.2 | <b>1.4</b>             | 23.9        | <b>48.2</b> | 10.4        | 32.4         | 9.2   |
|                 | SEUL     | 0.6               | 29.4 | 1.9                    | <b>23.6</b> | 48.0        | <b>11.4</b> | <b>28.3</b>  | 8.8   |
|                 | Original | 66.6              | 93.1 | 49.5                   | 82.5        | 50.7        | 12.3        | 23.3         | -     |
| GPT-Neo<br>2.7B | KUL      | 0.5               | 29.8 | 1.7                    | 23.0        | 49.9        | 10.1        | 30.1         | 8.6   |
|                 | SEUL     | 0.3               | 18.6 | <b>1.0</b>             | <b>14.6</b> | <b>50.2</b> | <b>10.9</b> | <b>26.7</b>  | 8.8   |
|                 | Original | 25.7              | 73.2 | 19.9                   | 58.2        | 56.9        | 13.2        | 9.0          | -     |
| LlaMa2<br>7B    | KUL      | 0.7               | 29.1 | 1.5                    | 22.3        | 52.1        | 9.7         | 21.0         | 7.1   |
|                 | SEUL     | 0.6               | 13.9 | <b>1.2</b>             | <b>12.3</b> | <b>53.6</b> | <b>10.4</b> | <b>16.3</b>  | 6.8   |
|                 | Original | 22.9              | 75.9 | 19.0                   | 63.9        | 62.5        | 14.1        | 9.1          | -     |
| Mistral<br>7B   | KUL      | 0.5               | 27.9 | 2.1                    | 23.6        | 53.3        | 13.3        | 19.8         | 7.0   |
|                 | SEUL     | 0.4               | 19.1 | <b>1.3</b>             | <b>20.9</b> | <b>55.7</b> | <b>12.8</b> | <b>17.0</b>  | 7.4   |

Table 1: Comparison results (in %) on forget set ( $d=32$ ), 8 classification datasets and 4 dialogue datasets. The best performance in comparable columns are highlighted in **bold**. “Avg.” denotes average scores.

Hellaswag (Zellers et al. 2019) to measure linguistic reasoning abilities, Winogrande (Sakaguchi et al. 2021), and COPA (Gordon, Kozareva, and Roemmele 2012) to measure commonsense reasoning abilities, and ARC-Easy (Clark et al. 2018), ARC-Challenge (Clark et al. 2018), Piqa (Bisk et al. 2020), MathQA (Amini et al. 2019), PubmedQA (Jin et al. 2019) benchmarks to measure scientific reasoning abilities) and 4 dialogue tasks (Wizard of Wikipedia (Dinan et al. 2019), Empathetic Dialogues (Rashkin et al. 2019), Blended Skill Talk (Smith et al. 2020), and Wizard of Internet (Komeili, Shuster, and Weston 2022)). These test sets are used to assess whether the general capabilities of language models are affected after unlearning.

**Evaluation Metrics.** We assess the performance of the methods from two perspectives: the effectiveness of unlearning and the maintenance of general performance. To evaluate the unlearning effectiveness on the forget set, we employ our proposed metrics S-EL<sub>n</sub> and S-MA, as well as EL<sub>n</sub> and MA, introduced in Section . As for the evaluation of maintaining performance, we use accuracy for the classification test sets and F1 and Perplexity (PPL) scores for dialogue tasks.

## Baselines and Implementation Details

**Baselines and Forgetting Threshold.** Our primary comparative analysis involves contrasting our SEUL with the approach proposed by Jang et al. (2023) (referred to as KUL). We follow their specified forgetting thresholds, based on EL<sub>10</sub> and MA representing the extraction likelihood and memorization accuracy of regular data. Importantly, these thresholds serve as empirical indicators of the targeted forgetting objectives for unlearning methods. For fair comparison, the results of SEUL reported are all achieved with our *online selection* method. We denote our results with *offline*

*annotation* as *Oracle* results and are compared in Table 4.

**Models and Training Details.** The unlearning experiments are conducted on the pre-trained language model GPT-Neo series (125M, 1.3B, and 2.7B), Llama2-7B and Mistral-7B. The learning rate for training is set to  $5 \times 10^{-5}$ , based on the selection from  $[2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}]$ . The variable denoting the number of forgetting instances, represented as  $d$ , is examined across the values  $d = 1, 2, 4, 8, 16, 32, 64, 128$ . Unless otherwise specified, the reported results in this paper are based on the  $d = 32$  setting. We adapt the global batch size during training to be the same as  $d$ , the number of forgetting instances, following Jang et al. (2023). Each setting is run 5 times and the reported results are the average of 5 different runs. All the models are trained with a single Nvidia GeForce RTX 3090.

## Experimental Results

### Main Results

The main comparative results of unlearning methods are presented in Table 1, we have the following observations:

- *Our SEUL generally exhibits superior effectiveness in unlearning sensitive information compared to the baseline.* As explained in Experimental Setup Section, we set the forgetting threshold consistent for both SEUL and KUL. This can be reflected by the results reported in Table 1 that they exhibit the same levels of EL<sub>10</sub> and MA. In this context, our SEUL demonstrates better scores in S-EL<sub>10</sub> and S-MA compared to KUL. Given that EL<sub>10</sub> and MA reflect general information leakage risk, and S-EL<sub>10</sub> and S-MA scores emphasize the risk of sensitive information leakage, these results indicate that our SEUL unlearning method offers better protection against sensitive information leakage.

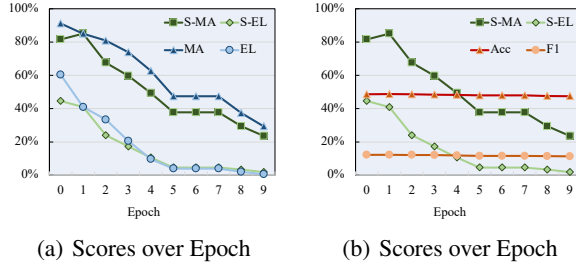


Figure 3: Unlearning GPT-Neo 1.3B: (a) S-MA, S-EL, MA, EL, and (b) Accuracy, F1 Scores over epochs.

- *Our SEUL demonstrates comparable results on classification datasets when compared to the KUL method, but it exhibits significantly better performance on dialogue datasets.* When examining the average accuracy scores on classification tasks, we observe that SEUL maintains an accuracy difference of  $0.1 - 0.3\%$  compared to the KUL method across three different GPT-Neo language model backbones (125M, 1.3B, and 2.7B). In contrast, when considering the average F1 and PPL scores on dialogue datasets, SEUL significantly outperforms KUL. This improvement may stem from that our fine-grained unlearning approach minimally impacts the generation performance of the language model, unlike KUL unlearning method that completely negate the loss function.

- *Our unlearning target does not result in longer training epochs.* Upon examining the average number of epochs required to reach the forgetting threshold, we observe that the average epochs for SEUL and KUL are similar. This suggests that our approach, which involves partially negating the loss function of pretraining, does not prolong the unlearning process compared to the fully reversed loss function employed by KUL.

- *Smaller language models exhibit less stability in maintaining performance.* When examining the unlearning results across different language models, ranging from small to large (125M to 2.7B), it becomes apparent that the 125M GPT-Neo model experiences the most significant performance drop when forgetting  $d = 32$  instances.

## Effectiveness Analysis of SEUL

**Performance Over Epochs.** We illustrate the evolution of scores across training epochs, focusing on Sensitive Extraction Likelihood (S-EL), Sensitive Memorization Accuracy (S-MA), EL, and MA scores in Fig. 3(a). Additionally, Fig. 3(b) displays the trends in S-EL, S-MA, average accuracy for classification datasets, and average F1 for dialogue datasets. In Fig. 3(a), the evaluation results of unlearning, with and without consideration of sensitive spans, demonstrate general consistency (i.e., S-MA and S-EL compared to MA and EL). Observing the results in Fig. 3(b), we note that as unlearning progresses (i.e., S-MA and S-EL degrade), the impact on both classification (Accuracy score) and dialogue (F1 score) performance is minimal. This observation underscores the effectiveness of our unlearning methods.

**Performance with Varying  $d$ .** The main results presented in Table 1 are derived from a forgetting set with a size of

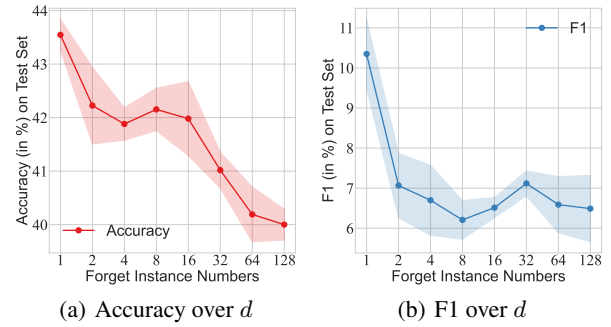


Figure 4: Unlearning GPT-Neo 125M: (a) Accuracy and (b) F1 Scores with varying  $d$  values.

|          |  |
|----------|--|
| Prefix   | Trump is the 45th  |
| Original | president of the United States. He is the first president to be elected in ... |
| KUL      | Nation to Target Black Voters Ahead of Election Day ...                        |
| SEUL     | president of the United States. The current president of the United States ... |

Table 2: Given a prefix, the suffixes generated by the original GPT-Neo 1.3B model, KUL and SEUL unlearned.

32. Here, we investigate the impact of unlearning based on different numbers of forgetting instances. The results for unlearning GPT-Neo 125M are shown in Fig. 4. Analyzing the trend in classification accuracy shown in Fig. 4(a), we observe that, while the overall performance drop is not substantial, there is a noticeable degrading trend as the number of forgetting instances increases. However, when assessing generation performance using the F1 score (Fig. 4(b)), we note that as  $n$  increases, the F1 scores remain more stable than the classification performance. Fig. 4(b) highlights the superiority of SEUL in maintaining generation performance, consistent with the observations discussed in Main Results Subsection based on Table 1.

**Stability under Adversarial Attacks.** As illustrated in Fig. 1, attackers may inject common sense or general knowledge into their original information and subsequently request unlearning. Under such attacks, full loss negation-based method (i.e., KUL (Jang et al. 2023)), may inadvertently forget the certain knowledge during unlearning. In contrast, our method demonstrates a capacity to withstand such attacks due to our selective unlearning. To simulate this adversarial scenario, we add additional common knowledge (e.g., “*Trump is the 45th president of the United States.*”) — information easily generated by the original language model — into every instances in forget dataset  $D_f$  and then conduct unlearning. We showcase the continuation results of models (with greedy decoding) when given the prefix “*Trump is the 45th*” in Table 2. The results in Table 2 demonstrate that KUL is unable to continue the commonsense sentence after unlearning, while our SEUL method can handle such



| Method             | Span Annotation   |
|--------------------|---|
| NER                | XXXX@XXXmail.com. * * Permission is hereby granted, free of charge, to any person obtaining a copy* of this software and associated documentation files (the Software ), to deal* in the Software without restriction, including without limitation the rights* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell* copies of the Software , ... |
| Online Selection   | XXXX@XXXmail.com. * * Permission is hereby granted, free of charge, to any person obtaining a copy* of this software and associated documentation files (the Software), to deal* in the Software without restriction, including without limitation the rights* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell* copies of the Software, ...   |
| Offline Annotation | XXXX@XXXmail.com. * * Permission is hereby granted, free of charge, to any person obtaining a copy* of this software and associated documentation files (the Software), to deal* in the Software without restriction, including without limitation the rights* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell* copies of the Software, ...   |

Table 3: Comparing annotated spans of NER, Online Selection, and LLM-based Offline Annotation.

| Models            | Forget Evaluation  |      | Avg. 8 Cla. | Avg. 4 Dia. |              | Epoch |
|-------------------|--------------------|------|-------------|-------------|--------------|-------|
|                   | S-EL <sub>10</sub> | S-MA | Acc(↑)      | F1(↑)       | PPL(↓)       | #     |
| SEUL              | 0.3                | 16.1 | 41.0        | <b>7.8</b>  | 179.8        | 16.2  |
| SEUL <sup>O</sup> | 0.3                | 14.6 | <b>41.5</b> | <b>7.8</b>  | <b>102.4</b> | 13.0  |

Table 4: Results (in %) for SEUL (trained on online selected data) and SEUL<sup>Oracle</sup> (trained on offline annotated data).

| Method  | Avg. Span # | Avg. Prop. | Cover (Human, ·) |
|---------|-------------|------------|------------------|
| Human   | 1.8         | 3.1%       | -                |
| NER     | 19.2        | 8.9%       | 5.3%             |
| Online  | 8.6         | 13.7%      | 72.0%            |
| Offline | 4.5         | 9.5%       | 75.1%            |

Table 5: Statistics of sensitive spans annotated by various methods. Avg. Span # refers to the average spans per instance, Avg. Prop. refers to the proportion of sensitive spans in the sequence. Cover (Human, ·) indicates the percentage of human-annotated spans covered by other annotations.

attacks. This underscores the superiority and robustness of our unlearning methods.

**Oracle Sensitive Span Based Unlearning.** It is noteworthy that our approach based on online selection emphasizes avoiding reliance on pre-determined sensitive or forgetting spans to produce a more general and cost-efficient solution. However, this inevitably introduces some inaccurately detected spans when evaluating with S-EL and S-MA based on ChatGPT-annotated spans. Therefore, we further explore unlearning based on offline annotated sensitive spans (denoted as SEUL<sup>Oracle</sup>) to assess the magnitude of this discrepancy. We stop the training of SEUL<sup>Oracle</sup> when its S-EL<sub>10</sub> S-MA scores are less or equal to SEUL, and then report the corresponding accuracy and F1 scores in Table 4. From the results in Table 4, we can see that SEUL<sup>Oracle</sup> shows a better capability in maintaining the performance of language model, thanks to its accurate annotation and less training epochs.

### Reliability of Sensitive Span Annotation

SEUL’s training and evaluation rely on automatic online or offline annotated sensitive spans. To ascertain the reliability of the annotation, we compare our automatic annotation with human annotation and Named Entity Recognition

(NER) based annotation. Specifically, we invite two annotators to do sensitivity annotation on 50 instances to be forgotten and then get 50 annotated instances regarded as ground truth. We also employ NER toolkit to annotate the same 50 instances where extracted entities are regarded as sensitive information. We have the following quantitative and qualitative analyses.

**Quantitative Comparison.** We conduct a statistical analysis of annotated sensitive spans derived by these methods and display results in Table 5. The results reveal that human annotators apply a more stringent criterion for sensitive annotation, leading to a smaller average number of sensitive spans per instance. In contrast, NER-based annotation labels more spans, but these spans exhibit a lower overlap with the spans annotated by human annotators. Our online sensitive selection method tends to label more spans while offline annotation achieves the highest matching with human results.

**Case Study.** We present a case study using the example in Table 3 to illustrate the annotation results of NER, Online Selection, and Offline Annotation. As observed, NER primarily extracts potential entities from the sequence, without specifically identifying sensitive information. Both Online Selection and Offline Annotation successfully annotate sensitive information, e.g. the email address (masked to avoid privacy issues). Nevertheless, it’s important to note that Online Selection may introduce additional spans that do not accurately represent sensitive information.

## Conclusion

In summary, this paper presents SEUL, a novel selective unlearning method for language models that minimizes negative impacts on model capabilities by focusing on specific sequence spans. It introduces specialized evaluation metrics, S-EL and S-MA, designed to assess the forgetting of sensitive information. The paper also proposes efficient automatic online and offline sensitive span annotation methods to support the overall unlearning framework. Overall, these contributions address concerns regarding the inadvertent retention of personal or sensitive information by neural models.

## Acknowledgments

The research described in this paper is partially supported by HK RGC GRF #14206324, The National Natural Science Foundation of China (Grant No. 62227808) and Shenzhen Science and Technology Program (Grant No. ZDSYS20210623091809029).

## References

- Amini, A.; Gabriel, S.; Lin, S.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2357–2367. Minneapolis, Minnesota: Association for Computational Linguistics.
- Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7432–7439.
- Bourtole, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159. IEEE.
- Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480. IEEE.
- Cettolo, M.; Niehues, J.; Stüker, S.; Bentivogli, L.; and Federico, M. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*.
- Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafford, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *ArXiv*, abs/1803.05457.
- Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.
- Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.
- Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.
- Golatkar, A.; Achille, A.; and Soatto, S. 2020a. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9304–9312.
- Golatkar, A.; Achille, A.; and Soatto, S. 2020b. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, 383–398. Springer.
- Gordon, A.; Kozareva, Z.; and Roemmele, M. 2012. SemEval-2012 Task 7: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 394–398. Montréal, Canada: Association for Computational Linguistics.
- Guo, C.; Goldstein, T.; Hannun, A.; and Van Der Maaten, L. 2019. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*.
- Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 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)*, 14389–14408. Association for Computational Linguistics.
- Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W.; and Lu, X. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2567–2577.
- Karasuyama, M.; and Takeuchi, I. 2009. Multiple incremental decremental learning of support vector machines. *Advances in neural information processing systems*, 22.
- Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.
- Komeili, M.; Shuster, K.; and Weston, J. 2022. Internet-Augmented Dialogue Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8460–8478. Dublin, Ireland: Association for Computational Linguistics.
- Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.
- Ling, Z.; Fang, Y.; Li, X.; Huang, Z.; Lee, M.; Memisevic, R.; and Su, H. 2023. Deductive Verification of Chain-of-Thought Reasoning. *arXiv preprint arXiv:2306.03872*.
- Liu, A.; Sap, M.; Lu, X.; Swayamdipta, S.; Bhagavatula, C.; Smith, N. A.; and Choi, Y. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.
- Lu, X.; Welleck, S.; Hessel, J.; Jiang, L.; Qin, L.; West, P.; Ammanabrolu, P.; and Choi, Y. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35: 27591–27609.
- Mehta, R.; Pal, S.; Singh, V.; and Ravi, S. N. 2022. Deep Unlearning via Randomized Conditionally Independent Hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10422–10431.
- Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and editing factual knowledge in gpt. *arXiv preprint arXiv:2202.05262*, 1.
- Patil, V.; Hase, P.; and Bansal, M. 2023. Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks. *arXiv preprint arXiv:2309.17410*.



- Rashkin, H.; Smith, E. M.; Li, M.; and Boureau, Y.-L. 2019. Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5370–5381. Florence, Italy: Association for Computational Linguistics.
- Romero, E.; Barrio, I.; and Belanche, L. 2007. Incremental and decremental learning for linear support vector machines. In *International Conference on Artificial Neural Networks*, 209–218. Springer.
- Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.
- Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K. R.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Smith, E. M.; Williamson, M.; Shuster, K.; Weston, J.; and Boureau, Y.-L. 2020. Can You Put it All Together: Evaluating Conversational Agents’ Ability to Blend Skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2021–2030. Online: Association for Computational Linguistics.
- Tirumala, K.; Markosyan, A.; Zettlemoyer, L.; and Aghajanyan, A. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35: 38274–38290.
- Tuggener, D.; Von Däniken, P.; Peetz, T.; and Cieliebak, M. 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 1235–1241.
- Wang, L.; Chen, T.; Yuan, W.; Zeng, X.; Wong, K.-F.; and Yin, H. 2023. KGA: A General Machine Unlearning Framework Based on Knowledge Gap Alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13264–13276. Toronto, Canada: Association for Computational Linguistics.
- Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; and Weston, J. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2204–2213.