Artificial Intelligence Review (2025) 58:90
https://doi.org/10.1007/s10462-024-11078-6

Digital forgetting in large language models: a survey of unlearning methods

Alberto Blanco-Justicia[1] · Najeeb Jebreel[1] · Benet Manzanares-Salor[1] · David Sánchez[1] · Josep Domingo-Ferrer[1] · Guillem Collell[2] · Kuan Eeik Tan[2]

Accepted: 13 December 2024 / Published online: 13 January 2025
© The Author(s) 2025

Abstract
Large language models (LLMs) have become the state of the art in natural language processing. The massive adoption of generative LLMs and the capabilities they have shown have prompted public concerns regarding their impact on the labor market, privacy, the use of copyrighted work, and how these models align with human ethics and the rule of law. As a response, new regulations are being pushed, which require developers and service providers to evaluate, monitor, and forestall or at least mitigate the risks posed by their models. One mitigation strategy is digital forgetting: given a model with undesirable knowledge or behavior, the goal is to obtain a new model where the detected issues are no longer present. Digital forgetting is usually enforced via machine unlearning techniques, which modify trained machine learning models for them to behave as models trained on a subset of the original training data. In this work, we describe the motivations and desirable properties of digital forgetting when applied to LLMs, and we survey recent works on machine unlearning. Specifically, we propose a taxonomy of unlearning methods based on the reach and depth of the modifications done on the models, we discuss and compare the effectiveness of machine unlearning methods for LLMs proposed so far, and we survey their evaluation. Finally, we describe open problems of machine unlearning applied to LLMs and we put forward recommendations for developers and practitioners.

Keywords Large language models · Machine unlearning · Privacy · Copyright · Trustworthy AI

1 Introduction

Large language models (LLMs) have become the state of the art in most if not all natural language processing (NLP) and natural language understanding (NLU) tasks. Since the publication of the transformer architecture by Vaswani et al (2017), several authors and com-

Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares-Salor, Josep Domingo-Ferrer, Guillem Collell and Kuan Eeik Tan contributed equally to this work.

Extended author information available on the last page of the article

panies have made use of (variations of) this architecture to tackle tasks such as translation, summarization, question answering, sentiment analysis, or text generation.

Whereas the original transformer consisted of an encoder-decoder architecture, variations of this have been introduced in subsequent works. Namely, encoder-only architectures such as BERT (Devlin et al 2018) –used mostly for text classification–, decoder-only architectures such as GPT (Radford et al 2018) – for text generation–, and encoder-decoder architectures, such as T5 (Raffel et al 2020) –for text-to-text processing, such as translation–.

The training of LLMs occurs in different phases. These phases include a self-supervised pre-training phase using text data collected from several sources (and sometimes uncurated), with tasks or training objectives that depend on the type of LLM being trained (*e.g.*, masked language modeling and next word prediction); a fine-tuning phase, where labeled data are used to train specific tasks, such as sentiment analysis, conversational text generation, code generation, or summarization; an additional optional round of fine-tuning using reinforcement learning from human feedback (RLHF), in which human annotators help fine-tune the model by providing feedback on the model responses; and a final phase for generative models, where specific *prompts* can be passed to the model to guide it and condition its future responses.

The announcement and publication of ChatGPT by OpenAI in November 2022 (among other large language, image, video, and audio generative models) released the capabilities of LLMs to the general public. Since then, several issues have been raised, mainly related to the *alignment* of such models with societal values and the rule of law. Such concerns have elicited political reaction at an international level, with governments and supranational entities reaching agreements such as the G7 Hiroshima AI Process (Group of Seven (G7) 2023), and adopting executive and legislative actions such as the President Biden's Executive Order on the Safe, Secure, and Trustworthy Development and Use of AI (Joseph R. Biden Jr. 2023), and the European Union's AI Act (European Parliament 2024).

These recommendations and regulations, along with privacy regulations like the European Union's General Data Protection Regulation (GDPR) (European Parliament and Council of the European Union, 2016), require developers and entities that deploy large (generative) models to document, monitor, inform about, and solve any potential risks during the development and operation of the models. Such risks include the impact of LLMs on several areas: the labor market, the right to privacy of individuals, copyright laws, the furthering of biases and discrimination, and the potential generation of harmful content, including content that could be used to damage people.

One proposed solution to these issues is digital forgetting, which includes machine unlearning techniques. Given a model with undesirable knowledge or behavior, the objective of digital forgetting is to obtain a new model where the detected issues are no longer present. However, effective digital forgetting mechanisms have to fulfill potentially conflicting requirements: (i) effectiveness of forgetting, that is, how well the new model has forgotten the undesired knowledge/behavior (either with formal guarantees or through empirical evaluation); (ii) performance retained by the model on the desirable tasks; and (iii) cost and scalability of the forgetting procedure.

1.1 Contributions and differences with other surveys

In this paper, we provide an up-to-date survey on digital forgetting and machine unlearning on LLMs. We first discuss and characterize the motivations, types and requirements of digital forgetting in LLMs. Then, we survey approaches to digital forgetting in LLMs and focus on machine unlearning techniques, which are the most common and effective methods. We have organized the surveyed methods according to the depth and reach of the modifications they perform on the models, from architectural changes to no changes at all, including methods that modify all or part of the model parameters. We also discuss the evaluation of unlearning methods in LLMs, including datasets, models, and metrics.

Most of the surveyed works were published in 2023 and early 2024. Specifically, we have analyzed and compared 22 unlearning methods, 28 datasets used for unlearning and evaluation, 31 datasets used for evaluation of retaining, and 15 models (plus size variations) used in the unlearning literature.

Finally, we provide a summary of the challenges of machine unlearning in LLMs and we state recommendations for practitioners on how to choose specific methods depending on their needs.

Related surveys on digital forgetting and machine unlearning, such as Qu et al (2023); Xu et al (2023); Nguyen et al (2022); Shaik et al (2023), have focused on the general ML case by considering models used to classify tabular or image data. However, natural language and LLMs have substantially distinctive qualities that warrant a dedicated study. As a consequence, and also looking at the large corpus of (so far unsurveyed) works on unlearning for LLMs published in the last 8 months, we believe that an up-to-date survey devoted specifically to this subject is much justified and needed.

1.2 Organization of this paper

This document is organized as follows. Section 2 describes the motivations, types, and desired properties of digital forgetting. Section 3 introduces the approaches to digital forgetting in LLMs, among which unlearning methodologies stand out as the state of the art. Section 4 provides a detailed taxonomy of machine unlearning methods for LLMs, and surveys and compares current approaches. Section 5 details datasets, models and metrics used for the evaluation of forgetting, retaining and runtime. Section 6 discusses challenges and recommendations in the area. Finally, Sect. 7 gathers concluding remarks.

2 Digital forgetting

Digital forgetting refers to procedures used to remove information from software systems. This term originated from privacy laws related to the right of erasure or the *right to be forgotten* (*e.g.* GDPR, Art. 17, (European Parliament and Council of the European Union, 2016)), which enables data subjects to request service providers to delete information related to them from their systems. However, forgetting can be extended to any request (by data subjects, regulators, or the entity managing the software system) to remove any information or unwanted behavior from a software system. In the case of LLMs, this translates to any modification in the model or the training procedures to remove any influence of specific training data or specific training procedures on the model's capabilities or behavior.

This section discusses the motivations for digital forgetting, including legal and ethical issues, the types of digital forgetting, and the desirable properties of forgetting procedures.

2.1 Motivations for digital forgetting

The need for digital forgetting stems from several ethical principles, national and supranational regulations, and codes of conduct. In general, regulations relating to the protection of the privacy of individuals (*e.g.*, the GDPR), the protection of intellectual property (*e.g.*, copyright laws), and the alignment with human values and the rule of law (Group of Seven (G7) 2023; Joseph R. Biden Jr. 2023; European Parliament 2024) motivate the research, discussion, and implementation of digital forgetting measures in software systems, including LLMs.

In the following, we categorize and discuss the reasons for implementing digital forgetting in LLMs.

2.1.1 Privacy

ML models, including LLMs, are trained on vast amounts of data, often obtained from open sources on the web. Misconfigured services sometimes include private data, like Personally Identifiable Information (PII) or business data, that can be indexed by search engines and freely accessed. These unintentionally public data may end up in the pre-training datasets of LLMs. Additionally, private data may be used to fine-tune models and teach them new downstream tasks. As an example, hospitals may fine-tune pre-trained models using patients' data to teach the models how to diagnose new patients.

General ML models have been shown to memorize and leak data from the training dataset, and this also holds for LLMs. Membership inference attacks (MIA) (Shokri et al 2017; Salem et al 2018; Carlini et al 2021) seek to determine whether some information was part of the training set of a trained model. The vulnerability of models to such attacks can be used to assess the risk of data leakage. Outlying data are especially at risk of being memorized and leaked, which in the case of personal data may lead to privacy disclosure (Smith et al 2023).

2.1.2 Copyright protection

Digital forgetting for copyright protection is closely related to privacy, as in both cases we do not want the generated text to include specific information. However, there is a key distinction: whereas in copyright protection the goal is to avoid leakage of verbatim text, in privacy protection one wants to avoid disclosing personal information no matter how it is expressed.

A model that can answer questions about an individual and whose answers contain sensitive details about them is infringing their right to privacy. However, the requirements of copyright protection are different. Copyright laws do not protect ideas, but the exact form in which they are expressed.[1] Furthermore, some specific items might be protected by trademark. Therefore, a model that can answer questions about some copyrighted work does not necessarily infringe copyright law unless verbatim fragments of the work are returned (and even in that case, the law includes provisions to lawfully quote protected works).

2.1.3 Model robustness

The whole LLM training pipeline includes pre-training from public data, fine-tuning with public, crowdsourced, or proprietary data, and possibly further fine-tuning with RLHF, which may also be public, crowdsourced, or closed. In all these phases of training, there is a possibility to process low-quality information. Datasets used for pre-training may include false, inconsistent, or outdated information. Datasets used for fine-tuning may be crowdsourced, which opens the door to malicious actors providing wrong information. RLHF depends on human annotators, who rank model outputs based on usefulness and safety. Rogue annotators may provide wrong information.

All these sources of outdated information, misinformation, outliers, noise, and malicious reporting may influence the learning process and thus produce underperforming models with potentially critical failures. Moreover, in the case of generative models, problems during training may cause the models to hallucinate, although this may be an intrinsic feature of generative LLMs caused by the random sampling of output tokens. Forgetting procedures may be used to correct some of these issues.

2.1.4 Alignment with human values

LLM pre-training datasets are compiled from diverse, often uncurated sources such as web pages, online encyclopedias, social media posts, and online book libraries. Some of these sources may contribute content misaligned with current societal values, including discriminatory language based on gender, race, ethnicity, sexual orientation, or age, often manifesting as stereotypes or prejudices. Notably, studies have identified strong correlations between pronouns (he/she) and different careers. Furthermore, these sources may include hateful, violent, or harmful speech. Pre-training models on such data may not only perpetuate these biased and harmful behaviors but even amplify them in some cases. Generative models may also learn tasks that are not aligned with the predefined goals and which could turn out to be harmful, such as how to build weapons.

ML models, including LLMs, must not only protect individual privacy rights, as discussed above, but also adhere to ethical values and principles, taking as a reference the Universal Declaration of Human Rights (United Nations 1948), and also including laws and social norms. Alignment with principles such as non-discrimination, fairness, benevolence, and non-maleficence is of utmost importance. In addition, the EU Guidelines for Trustworthy AI (European Commission 2019) require that the machine learning models being developed be always under human control and supervision, which means that any deviation from such principles should be addressed (or addressable).

[1] WIPO Copyright Treaty, Article 2: Scope of Copyright Protection – Copyright protection extends to expressions and not to ideas, procedures, methods of operation or mathematical concepts as such. https://www.wipo.int/wipolex/en/text/295166

Therefore, alignment with ethical values may necessitate forgetting procedures aimed at identifying and eliminating sources of discriminatory or harmful behavior.

2.2 Types of digital forgetting

In this section, we discuss the types of digital forgetting requests that can be found in previous surveys on general machine unlearning (Qu et al 2023; Xu et al 2023; Nguyen et al 2022; Shaik et al 2023). These surveys include forgetting requests for items, features/concepts, classes, and tasks. Figure 1 visually summarizes these types of forgetting.

While discussing these types of requests, we will describe how they apply to the case of natural language processing and language models.

Item removal requests. In the general ML case, item removal requests are concerned with one or more specific data points or samples in the training data used to train the model. Such requests are straightforward for models dealing with tabular or image data, where data points are precisely defined. In the realm of tabular data, each row within a table is considered a data point, with one of the attributes serving as the class label for classification or regression. Likewise, in computer vision tasks, individual images are the designated data points.

However, the distinction between items becomes less evident in NLP tasks and LLMs. Whereas one might consider each token in a text dataset as an individual data point, these tokens often lack meaning of their own. Consequently, in NLP data points can consist of entire sentences or even whole documents, as the separation between meaningful units is less neat than in tabular or image-based data scenarios.

Feature or concept removal requests. Requests for feature removal are concerned with specific attributes about data points. This is well defined for tabular data, where attributes correspond to one or more columns of the tables.

For images, the definition of what is a feature is less clear-cut, as features of images may correspond to individual pixels, to information automatically extracted by the models to achieve some predictions, or even be the classification targets in other cases. For example, a model that classifies images of dogs and cats may internally extract features that represent the shape of the ears to make a decision. Therefore, a feature is information about a data point, which may or may not be common to other data points and that serves to partially describe these data points (even if these descriptions are only meaningful within the models).

Record	Attribute 1	Attribute 2	Attribute 3	Task 1	Task 2
				Class Label	Class Label
1	a	w	l	Positive	Technology
2	b	x	m	Negative	Health
3	c	y	n	Positive	Finance
4	d	z	o	Negative	Environment

Item removal   Feature removal   Class removal   Task removal

Fig. 1 Types of digital forgetting

In the case of NLP, features may be defined as words or tokens with strong connections[2] to other specific words or tokens of interest (such as an identity). This information may be spread across different sentences and documents. An example, related to privacy protection, could be that of a request in which all information related to a data subject is required to be removed from a model. Where features apply to a broad range of items, we refer to them as concepts.

Class removal requests. Class removal requests consist in removing all information about a whole class from a model. These requests are quite natural for models used to identify people. For example, in facial recognition, each of the classes corresponds to a single individual, and data points in the class are images of said individual. Thus, removing one class amounts to making the model unable to identify a person that the model could previously identify.

Class removal requests can be directly translated to NLP classification tasks, such as sentiment analysis. In this case, removing a class requires making the model unable to recognize sentences of a given sentiment, either by removing all sentences that belong to the class, or by making the model classify them as a different class.

However, for generative models, each class corresponds to a word or token in the vocabulary of the model, and removing it could directly impact the capabilities of the model.

Task removal requests. As described in previous sections, LLMs are pre-trained on large text datasets with generic tasks, such as next-word prediction or masked language modeling. After pre-training, models are fine-tuned to teach them different tasks, such as summarization, sentiment analysis, code writing, conversational text generation, etc. Task removal requests attempt to make the fine-tuned LLMs forget one or more of the tasks that the model has been taught to perform.

2.3 Requirements of digital forgetting

Regardless of the reason or type of forgetting request, we can define a series of general requirements ensuring that the forgetting procedure is carried out correctly and the resulting model still performs adequately. This is the purpose of this section, but first, we introduce some preliminary definitions.

A dataset D consists of samples $\{x_i, y_i\}_{i=1}^{N}$, where N is the size of the dataset, x_i are token sequences, and y_i the true labels. Note that in self-supervised training, used for most pre-training tasks in LLMs, the labels y_i do not need to be explicitly defined. For example, in next-token prediction y_i is the token immediately following the sequence x_i in the text. However, during fine-tuning, labels are expected to be explicitly provided. A forget dataset $D_f \subset D$ is the set of samples to be forgotten. The retain set is defined as $D_r = D \setminus D_f$.

A learning algorithm $A(D)$ is a probabilistic algorithm that outputs a model given a training dataset D. Due to the probabilistic nature of $A(\cdot)$, we cannot ensure that running the learning algorithm twice on the same dataset will return the same model. However, we can define a probability distribution $P(A(D))$ over all possible models returned by the learning algorithm when trained on the same dataset D. Additionally, we can define $Dist(\cdot, \cdot)$ as the distance between two probability distributions. An example of such a distance is the Kullback–Leibler (KL) divergence.

[2] A strong connection between tokens could be interpreted as closeness in the embedding space, or as strong attention values between them.

A forgetting algorithm $F(D_f, A(D))$ is a (probabilistic) algorithm that returns a model in which the influence of the samples in D_f has been removed from *A*(*D*). Definitions of unlearning by Nguyen et al (2022); Xu et al (2023) include *D* as a parameter of the unlearning mechanism (as in $F(D, D_f, A(D))$), but access to *D* is not always feasible. For example, a service provider that leverages a foundation model such as Llama2 to offer some service (possibly after some additional fine-tuning) does not have access to the data used by the party that conducted the pre-training. However, the service provider may still be required by law to fulfill forgetting requests. For the sake of generality, we will abstain from including *D* as a parameter for forgetting, although *D* may be used in some procedures.

When implementing digital forgetting in ML models (and in LLMs in particular), the following requirements should be taken into consideration.

Forgetting guarantees. An essential requirement for any forgetting procedure, whether in the conventional scenario of search engines or with large language models, lies in the ability to demonstrate the fulfillment of a forgetting request, particularly when such fulfillment is a legal obligation. A forgetting guarantee serves as a theoretical proof, offering assurance that the content associated with a forgetting request has been forgotten, accompanied by a level of certainty. This ensures a transparent and accountable process in meeting legal requirements and reinforces the credibility of the forgetting mechanism. Nguyen et al (2022) refer to two levels of guarantees, *exact* and *approximate*. Xu et al (2023) further expand approximate forgetting into *strong* and *weak* forgetting, where strong forgetting is equivalent to the definition of approximate forgetting by Nguyen et al (2022), and weak forgetting only applies to the outputs of the model. However, most of the literature on unlearning mechanisms provides no provable guarantees, relying instead on empirical evaluation or auditing.

- *Exact forgetting.* A forgetting algorithm $F(D_f, A(D))$ provides an *exact* forgetting guarantee if

$$Dist(P(F(D_f, A(D))), P(A(D \setminus D_f)) = 0,$$

where $P(\cdot)$ denotes probability distribution. From this definition, we can conclude that retraining from scratch on the retain set D_r is a straightforward approach to exact unlearning, since $P(A(D_r)) = P(A(D \setminus D_f))$

- *Approximate forgetting.* A forgetting algorithm $F(D_f, A(D))$ provides an *approximate* forgetting guarantee, if

$$Dist(P(F(D_f, A(D))), P(A(D \setminus D_f)) \leq t,$$

for an acceptable threshold *t*. Nguyen et al provide a definition for ϵ-certified forgetting, inspired by differential privacy. Given $\epsilon > 0$ and a sample $z \in D$,

$$e^{-\epsilon} \leq \frac{\Pr(F(z, A(D)))}{\Pr(A(D \setminus z))} \leq e^{\epsilon}.$$

- *Weak/No guarantees.* The forgetting guarantees described above may not be attainable in all cases, because achieving them or computing them is unfeasible. In these cases, we can refer to empirical evaluation or auditing to provide a level of risk reduction (that is, how much the risk of the model remembering the undesired item has been reduced). These evaluations will depend on the type of forgetting request that needs to be dealt with. When requests are related to privacy or copyright protection, the difference in membership inference vulnerabilities can serve as a measure of risk reduction. If the forgetting request involves corrections of biases in the models, fairness metrics can be used. Finally, if some task needs to be forgotten, specific task benchmarks can be used. Refer to Sect. 5.3 for different evaluation mechanisms.

Retaining of performance. Whatever the method used for forgetting, the resulting models should retain much of the performance of the original model with respect to the same metrics and benchmarks. Note that if a given task is removed from the resulting model, some specific benchmarks may no longer be applicable. Xu et al (2023) consider two performance metrics for classifiers, namely consistency and accuracy. Consistency is defined as the level of agreement between a model resulting from a forgetting procedure and a model trained only on the retain dataset. Accuracy is defined in a standard way, by evaluating the model resulting from a forgetting procedure on the retain dataset. While these metrics may not be directly applicable to generative models, other metrics, such as perplexity,[3] could be used to compare unlearned and retrained models.

Runtime and scalability. When serving forgetting requests, especially if triggered by privacy concerns or by the application of the right to be forgotten, it is important that the forgetting procedure can be executed promptly. Article 17 of the GDPR requires in paragraph 1 that "the controller shall have the obligation to erase personal data without undue delay"; additionally, paragraph 2 indicates that such procedures should be carried out "taking account of available technology and the cost of implementation". Thus, it is important that forgetting algorithms can be executed in a timely manner, so that no personal information is accessible for longer than needed (to minimize potential harm to individuals). Other types of forgetting requests, such as those seeking to delete copyrighted material, correct biases or remove tasks may not be so urgent.

A different, but related property is scalability, meaning how many forgetting requests can be processed simultaneously and how that affects the runtime and the utility of both the resulting model and the forgetting procedure.

3 Approaches to digital forgetting in LLMs

We next delineate the main approaches to digital forgetting in LLMs. These approaches (as illustrated in Fig. 2) are each suitable for different phases of the LLM pipeline, going from data pre-processing and privacy-preserving model pre-training, which are suitable approaches when models can be retrained from scratch, to machine unlearning, prompt engineering and post-processing

[3] Perplexity is an information-theoretic measure of uncertainty in the value of a sample from a discrete probability distribution. In language models, perplexity measures how well a language model predicts a text sample. It is calculated as the average number of bits per word a model needs to represent the sample.

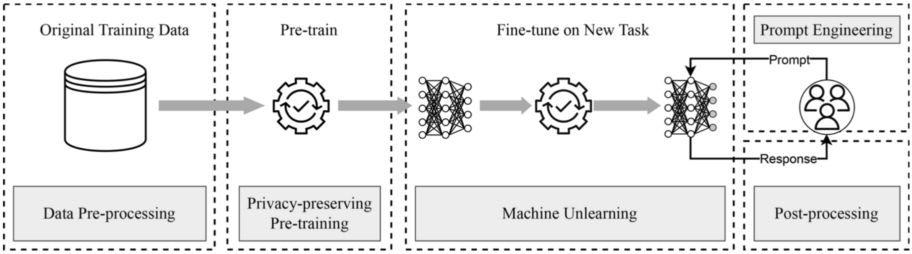

Fig. 2 Pipeline for training (top) and forgetting (bottom) in large language models

Data pre-processing and model retraining. Carefully choosing the data to be included in the pre-training and fine-tuning phases is a sensible and recommended approach to prevent any unwanted behavior from the models, be it from a privacy, a copyright, or an alignment perspective.

A second potential approach to limit the amount of private information in the training text is to perform text anonymization, also called text redaction or sanitization. Traditionally, redaction has been manually carried out by human experts. However, with the improvement of artificial intelligence mechanisms, some automated approaches have been proposed. Most approaches are based on named-entity recognition (either rule-based or ML-based) (Lison et al 2021). More more sophisticated approaches assess the sensitiveness of terms by leveraging the information theory (Sánchez and Batet 2016, 2017), or the semantics embedded in language models (Hassan et al 2019, 2023). In both types of approaches, sensitive items in the text are identified and then either removed or generalized.

A third approach based on data pre-processing is deduplication (Kandpal et al 2022). This consists in finding replications of the same text within the training corpus and deleting any duplicates. What constitutes a duplicate can be parameterized in terms of length. Duplicate documents tend to be more vulnerable to membership inference attacks, as shown by Carlini et al (2021); Nasr et al (2023). Thus, deduplication can be an effective mechanism against memorization (which may lead to privacy, copyright, robustness, and alignment issues).

According to the properties described above, retraining models that exclude the data to be forgotten yields an exact forgetting guarantee. However, the cost in time and resources of retraining models from scratch makes this option unattractive in the short term, especially if forgetting requests have to be executed urgently.

Privacy-preserving model pre-training. Using privacy-preserving ML mechanisms may limit the influence of any single data point on the model. In this case, instead of protecting the data, we use some training mechanism that ensures privacy. Mechanisms such as differentially private stochastic gradient descent (DP-SGD) (Abadi et al 2016) and private aggregation of teacher ensembles (PATE) (Papernot et al 2018) have been used in the last years to train privacy-preserving machine learning models.

As in the previous case, this approach will provide not only exact guarantees of unlearning, but provable guarantees of privacy. The application of these methods, however, will in most cases degrade the utility of the resulting models, and the resources and time needed for model training can be even higher than for retraining from scratch.

Machine unlearning. Given the high cost and time required to train LLMs, retraining them from scratch to eliminate undesirable behaviors is often an impractical endeavor. Currently, there is a growing trend in the literature to adopt the unlearning approach as an efficient means for digital forgetting in LLMs. Methods that attempt to remove undesirable knowledge or behaviors from models that have already undergone pre-training (and maybe also fine-tuning) without retraining those models from scratch are called *machine unlearning* mechanisms. These mechanisms rely on further fine-tuning, often with adversarial objectives, on the identification of parameters that are correlated with unwanted information and their modification, and on parameter arithmetic. Sections 4.1 and 4.2 below cover machine unlearning mechanisms of this kind.

These methods offer a wide range of forgetting guarantees, utility levels of the resulting models, and implementation costs.

Prompt engineering. Specific prompts to trained LLMs can be used to further steer the models' behavior after fine-tuning. These methods do not cause any changes to the model parameters and can in some cases be bypassed by inputting contradicting prompts. However, carefully crafted prompts inserted at the beginning of a conversation with a generative model can prevent it from generating private, biased, or harmful information. For example, Üstün et al (2024) use the prompt (preamble) *Does the following request contain harmful, unethical, racist, sexist, toxic, dangerous, offensive or illegal content or intent? If yes, explain that you do not engage in these types of requests.* They find that their model rejects up to 88% of such requests. Section 4.4 describes methods that use prompt engineering for digital forgetting.

Post-processing. Other technical measures that can be applied to models accessible only through an API are post-processing or filtering. After the models have generated an output and before serving it to the user, the service provider can analyze the output to search for unwanted generations, possibly using other LLM-based classifiers. Other approaches use a memory of unwanted responses to identify and filter out such generations. Section 4.4 includes some of these approaches.

4 Survey on unlearning in LLMs

As discussed in Sect. 3, unlearning is the most general way to efficiently eliminate undesirable or to-be-forgotten knowledge from LLMs without the need for full retraining. In this section, we conduct a comprehensive survey of unlearning methods applicable to LLMs, and we propose a fine-grained classification of methods into four primary categories: global weight modification, local weight modification, architecture modification, and input/output modification methods. This classification is predicated on the reach and depth of the alterations performed on the target models.

Global weight modification methods include those that have the potential to alter all model parameters of the target model during the unlearning process. Conversely, local weight modification methods are restricted to modifying a specific subset of parameters. Architecture modification methods do not modify the parameters of the model, but introduce additional trainable layers into the model's architecture. Figure 3 shows a conceptual illustration of the modifications associated with weight and architecture changes.

Finally, input/output modification methods do not make any changes to the model's architecture or parameters, and function exclusively by manipulating the inputs and outputs of the model. We further divide these primary categories based on how unlearning is performed.

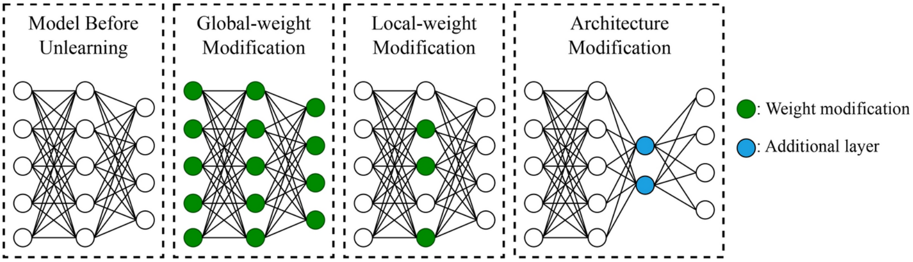

Fig. 3 Conceptual illustration of weight/architecture modification

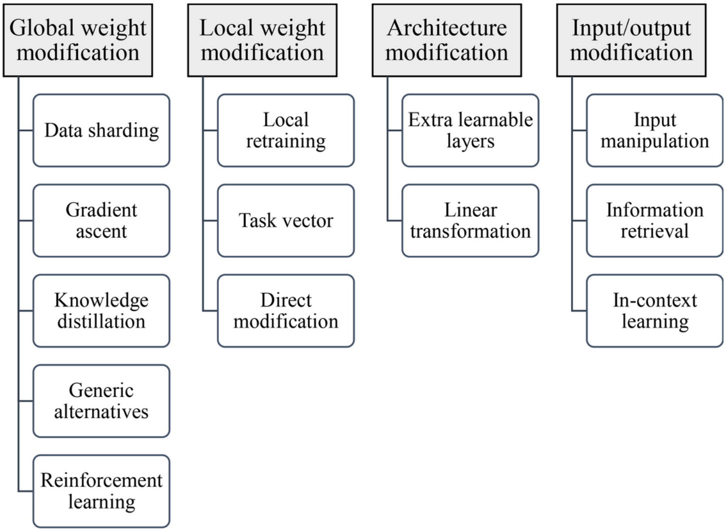

Fig. 4 Taxonomy of unlearning methods in LLMs

Figure 4 illustrates our proposed taxonomy of unlearning methods for LLMs, to be used as a framework for this survey.

4.1 Global weight modification

In these methods (Bourtoule et al 2021; Jang et al 2022), every parameter of the model is subject to modification during the unlearning process. Whereas global weight modification methods offer a stronger unlearning guarantee compared to the other approaches, they often entail substantial computational and time overheads, which renders them impractical for LLMs in many cases.

4.1.1 Data sharding

This approach typically entails dividing the training data into multiple disjoint shards, each corresponding to a subset of the overall data, and training a separate model for each shard

(Bourtoule et al 2021; Liu and Kalinli 2023). These individual models can then be leveraged to effectively remove data whose unlearning has been requested.

Bourtoule et al (2021) introduce SISA (Sharded, Isolated, Sliced, and Aggregated training) as a generic exact unlearning framework.

As depicted in Fig. 5, the training dataset is divided into S non-overlapping shards, each containing R slices. For each shard, an independent model is trained by processing the data slice by slice, saving a model checkpoint after each slice has been processed. During inference, the predicted label is obtained by aggregating the individual predictions of each model, such as in ensemble methods. When an unlearning request is received, the shard and slice where the data point is located are identified. The data point (typically a sequence of tokens in NLP tasks) is then removed from the this slice, and the model is retrained starting from the last unaffected checkpoint onward. For instance, in Fig. 5, retraining would commence from the checkpoint trained on $Slice_{2,1}$, which by construction ensures the model forgets the data point to be unlearned. The main advantage of SISA is that it provides an exact unlearning guarantee. This method can be applied to a wide range of ML tasks and model architectures, including LLMs. However, SISA is not very practical for LLMs due to the high computational/memory cost associated with model and checkpoint saving, retraining, and inference. Kadhe et al (2023) found that SISA can reduce fairness in LLMs and adapted a post-processing fairness improvement technique to make it fairer.

Liu and Kalinli (2023) propose the *leave-one-out (LOO) ensemble* method to reduce retraining cost and maintain utility at inference time. This method requires a base "student" model trained on the entire dataset and a collection of M "teacher" models trained

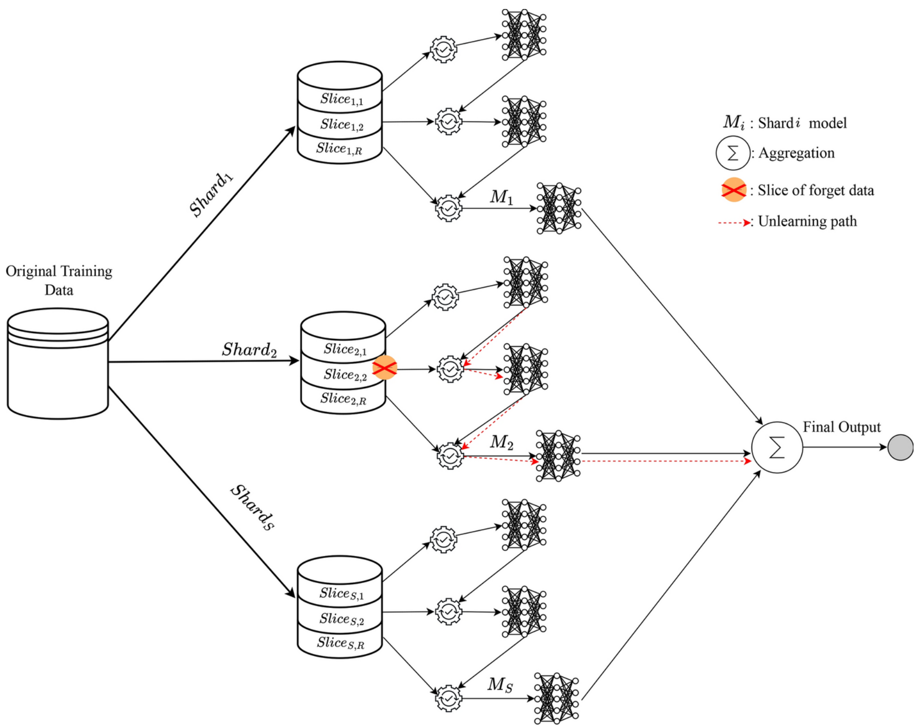

Fig. 5 Framework of SISA. Forgetting request processing in red

on disjoint subsets of the training dataset. Upon receiving a forget request, a prediction on the to-be-forgotten sequences is produced using the $M-1$ models that do not include the unwanted knowledge. The student model is then fine-tuned on the obtained soft labels. While offering improved utility compared to SISA, this method's unlearning guarantee for the base model is approximate, and there is no guarantee for the teacher models. Moreover, the process of predicting soft labels, aggregating them, and then fine-tuning the parameters of the base model incurs significant computational overheads, particularly with LLMs.

4.1.2 Gradient ascent

Methods under this approach aim to move the model away from undesirable knowledge by fine-tuning all model parameters to maximize loss on the related target tokens.

Given a set of target token sequences representing sensitive information, the method by Jang et al (2022) runs a number of optimization steps negating the original training loss function for those token sequences, therefore moving away from the minimum loss with respect to such sequences.

The authors found that unlearning many samples at once substantially degrades the performance of LLMs, and unlearning them sequentially can mitigate this degradation.

Gradient ascent (GA) and its variants only require the to-be-forgotten data and sometimes enhance the generalization capabilities of the model, as observed by Yoon et al (2023). However, GA can cause the model to lose its understanding of the language (Eldan and Russinovich 2023). Furthermore, the success of unlearning depends on the specific target data and the domain of the to-be-forgotten data (Smith et al 2023).

To preserve the generation capability of LLMs, SeUL (Wang et al 2024) applies GA (Jang et al 2022) to specific sensitive spans within the sequence instead of the entire sequence. The sensitive spans are selected using two annotation methods: an online method which assumes a token to be sensitive if it has a high perplexity score, and an offline method which employs the in-context learning capabilities of LLMs to annotate such sensitive spans. While this method allows for more utility-preserving, focused, and efficient unlearning compared to GA (Jang et al 2022), the annotation methods may lack precision in identifying sensitive tokens.

Yao et al (2023) observed that only applying GA as Jang et al (2022) do is insufficient to effectively unlearn unwanted (mis)behaviors (*e.g.*, harmful responses and hallucinations) and that preserving a good retain performance is harder than unlearning, because the retain performance greatly depends on the format of the retain data. Based on these observations, the authors propose an unlearning method that minimizes three weighted loss functions. The method involves updating the LLM during unlearning by jointly (1) applying GA on forget samples, (2) forcing random outputs on forget samples, and (3) minimizing the KL divergence between predictions of the original and unlearned models on retain samples. While the method provides a better trade-off between unlearning and retaining utility, it requires a large number of training epochs to unlearn forget data and maintain utility simultaneously.

4.1.3 Knowledge distillation

Methods under this approach treat the unlearned model as a student model that aims to mimic the behavior of a teacher model with the desired behavior.

Wang et al (2023) propose the Knowledge Gap Alignment (KGA) method as an unlearning technique for LLMs. KGA utilizes training data, data to be forgotten, and external data for unlearning to produce an updated model that exhibits a similar behavior on forgotten data as on unseen data while retaining utility on the remaining data. This is achieved by aligning the "knowledge gap," which refers to the difference in prediction distributions between models trained with different data. Aligning the unlearned model's behavior on the forget data with unseen data is achieved by minimizing the difference in the distributions of the unlearned model predictions on the forget data and the original model predictions on the unseen data. The authors use the KL divergence to measure this difference. The main advantage of this method lies in its generic nature, which allows it to be applied to various models and tasks. However, the need to train two identical models and then fine-tune all model parameters may limit its practicality and efficiency when applied to LLMs. Additionally, the unlearning process requires the training data, the data to be forgotten, and other external data with no overlapping with the training data. The utility of the unlearned model is highly dependent on the sizes of the data to be forgotten and the external data.

4.1.4 Generic alternatives

This approach aims to achieve unlearning by fine-tuning all the model parameters to predict generic alternatives instead of the to-be-forgotten tokens.

Eldan and Russinovich (2023) propose an unlearning method for erasing source-specific target data from LLMs, namely the Harry Potter books. Their three-step method involves:

- Token identification through reinforced modeling. This process involves creating an augmented model with a deeper understanding of the content that needs to be unlearned. This is done by further fine-tuning the original model on the target data. Tokens with significantly increased probability are identified, indicating content-related tokens that should be avoided during generation.
- Expression replacement. Distinctive expressions in the target data are replaced by generic alternatives. The authors used GPT-4 to generate those alternatives automatically. This helps approximate the next-token predictions of a model that has not seen the target data.
- Fine-tuning. Armed with these alternative labels, the model undergoes fine-tuning to erase the original text from the model's memory and provides a plausible alternative when prompted with its context.The authors demonstrated that the unlearned model no longer generates or recalls content related to the target data, while its performance on common benchmarks remains nearly unaffected. This method provides plausible alternatives to unlearned tokens, which is important for maintaining the retain utility in many scenarios. However, Shi et al (2023) found that models that underwent unlearning with this approach could still output related copyrighted content. Another limitation is that this approach relies on replacing unique terms with their generic alternatives, which is challenging when extending it to non-fiction content.

4.1.5 Reinforcement learning from human feedback (RLHF)

RLHF involves leveraging human feedback to guide the model's learning process. It combines reinforcement learning (RL) with human feedback to teach the model to generate text that aligns better with human preferences and intentions.

Lu et al (2022) present Quark, which considers the task of unlearning undesired behaviors of a target LLM by fine-tuning the model on signals of what *not* to do. Quark starts with a pre-trained LLM, initial training prompts, and a reward function to initialize a pool of examples. Then, it alternates between quantization, learning, and exploration. In quantization, the pool of examples is sorted according to the reward function and partitioned into quantiles. For learning, the model is fine-tuned on the quantized data pool using a standard language modeling objective and a KL penalty. During exploration, new generations are added to the data pool by sampling from the model, which is conditioned on the tokens with the higher rewards. The objective of the three-step process above is to teach the model to generate texts of varying quality with respect to the rewards. At inference time, the sampling is conditioned with the best reward token to steer toward desirable generations. Quark is one of the well-known state-of-the-art controllable text generation methods that effectively aligns the model output with human expectations (Daheim et al 2023). However, in addition to its computational cost significantly increasing as the data pool size increases, the model may still retain the sensitive information on its parameters.

4.2 Local weight modification

Unlike the previous methods, which were allowed to modify all model parameters, methods in this category are restricted to a certain subset of model weights during unlearning.

4.2.1 Local retraining

This approach employs a selective retraining strategy, where only the model parameters relevant to the unlearning targets are optimized, leaving the rest unaltered.

Yu et al (2023) introduce a gradient-based debiasing method called Partitioned Contrastive Gradient Unlearning (PCGU). PCGU systematically identifies and retrains specific model weights responsible for biased behavior. The approach involves computing the gradients for contrastive sentence pairs. The contrastive sentence pairs are pairs of sentences where only a key demographic attribute is altered, for example, changing a gendered pronoun. The differences in the computed gradients are then ranked to select which weights in the model contribute the most to the biased behavior of the model, and these weights are subsequently retrained. The authors demonstrate that the method is effective both in mitigating bias for the gender-profession domain as well as in generalizing these influences to other unseen domains. However, the work only addresses gender bias in masked LLMs and it remains uncertain whether the proposed method can be generalized to other kinds of LLMs and more complex social biases like race and social class discrimination.

4.2.2 Task vector

These methods build on the idea that an LLM can be decomposed into task-specific vectors. Then the vector of the task to be forgotten can be eliminated to achieve unlearning.

Ilharco et al (2022) introduce a novel paradigm for steering neural network behavior based on task vectors. A task vector τ is defined as the difference between the parameters of a model θ^t_{ft} fine-tuned on a specific task *t* and those of the corresponding pre-trained model θ_{pre} (a model not yet trained on task *t*). Employing τ allows for selective modification of the model's behavior for task *t* without significant impact on other tasks. This is done via the negation and addition arithmetic operations: negating τ allows forgetting knowledge related to task *t* while adding it improves the model's performance on *t*. Figure 6 illustrates how a task vector τ is obtained and how it is used for task forgetting via negation.

The authors show that this method can be used to improve the performance of the model on multiple tasks by adding their vectors. However, it cannot handle forgetting discrete facts or specific token sequences, which makes it more suitable for broader task-based modifications than for fine-grained ones.

By drawing inspiration from human cognitive processes and the task arithmetic in Ilharco et al (2022), the authors of Ni et al (2023) propose a paradigm of knowledge updating called F-Learning (Forgetting before Learning). Specifically, the target model is first fine-tuned with the knowledge to be unlearned. Then, the difference of the parameters of the target and the fine-tuned models is subtracted from the parameters of the target model. This process is termed "old knowledge forgetting." Finally, the unlearned model is fine-tuned with new knowledge, constituting the "new knowledge learning" stage. However, as it happened with Ilharco et al (2022), this method is not suitable for forgetting individual facts or sequences of tokens.

4.2.3 Direct modification

Instead of gradient optimization of the original or additional parameters, this approach locates and directly modifies the relevant parameters or neurons to achieve unlearning.

DEPN (Wu et al 2023) assumes that private information, such as usernames and contact information, is encoded in specific neurons of the target LLMs. Based on that assumption, the method first locates these neurons by using the integrated gradient method by Sundararajan et al (2017) and then sets their activations to zero to eliminate their influence on the output. A privacy neuron aggregator is also introduced to handle multiple unlearning requests. An analysis of the relationship between privacy neurons and model memorization was also performed. The authors found that model size, training time, and frequency of

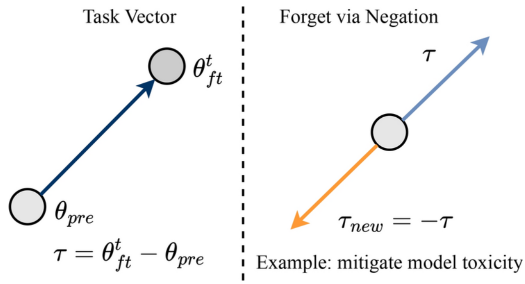

Fig. 6 Illustration of task vectors and the negation arithmetic operations for task forgetting

occurrence of private data are all factors that have an impact on model memorization. As the model memorization of private data deepens, the aggregation of privacy-related neurons associated with those data becomes more obvious. This method is efficient, as it only requires the forget set without fine-tuning. However, as the amount of forget data increases, the utility of the model drops significantly because more neurons are deactivated. Also, the authors found that including too many instances in a batch reduces the effect of forgetting. Besides, the method evaluation on forgetting private information was limited to names and phone numbers, due to the limited availability of datasets with a wide range of private data. The authors acknowledge the necessity of expanding their dataset to improve the effectiveness and relevance of their method.

Pochinkov and Schoots (2023) found that both feed-forward and attention neurons in LLMs are task-specialized, and that removing certain neurons from them significantly decreases performance on the forget task while hardly affecting performance on the other tasks. Based on that, they introduce a *selective pruning* method for identifying and removing neurons from an LLM that are related to a certain capability like coding or toxic generation. Neurons are removed based on their relative importance on a targeted capability compared to the overall model performance. This method is compute- and data-efficient in the identification and removal of task-specialized neurons. It can also be applied to remove other harmful skills, such as toxic generation and manipulation. However, the method requires a task-representative dataset and it is only effective for capabilities directly captured by these datasets. Besides, its effectiveness depends on the separability of the capabilities of an LLM. This method becomes less effective with models like Pythia (trained without dropout) and on smaller LLMs.

4.3 Architecture modification

In addition to the computational burden, altering all or a substantial amount of the target model's weights may degrade the model utility on the remaining data. To tackle these issues, methods based on architecture modification add extra learnable parameters within the model or use linear transformation in order to achieve efficient and utility-preserving unlearning (Chen and Yang 2023; Limisiewicz and Mareček 2022).

4.3.1 Extra learnable layers

EUL (Chen and Yang 2023) integrates an additional unlearning layer into transformer structures after the feed-forward networks. For each forgetting request, an unlearning layer is individually trained using a selective teacher-student objective. During the training of this layer, the original parameters are frozen, and a joint unlearning objective is used. The design of this joint objective is such that it compels the unlearning layer to diverge from the original model's predictions on the forget data, while following it on the remaining data. A fusion mechanism is also introduced to handle sequential forgetting requests. While this method maintains the model utility on the retain set, it has its limitations. It does not constitute complete forgetting, as removing the additional layers causes the old model to revert to its original behavior on the data to be forgotten, thus contradicting privacy laws. Also, training an additional layer is required to forget every unlearning request, which might not be scalable.

Kumar et al (2022) propose two variants of SISA to improve its efficiency on LLMs: SISA-FC and SISA-A. SISA-FC starts from a base model, pre-trained on a generic text corpus, and then adds fully connected (FC) layers on top of it. Only the parameters from the added layers are updated during retraining. This approach minimizes the overall retraining time, as backpropagation of gradients occurs only in the final layers, and storage requirements are reduced to the weights of these additional parameters. However, the utility of the model will be severely affected when compared to fine-tuning the entire model. SISA-A tries to address this by training the model using the parameter-efficient adapter method from Houlsby et al (2019), which injects learnable modules into the encoder blocks of the transformer. The methods were evaluated on classification tasks with the BERT model with no evaluation for forgetting performance. While SISA-A preserves the classification utility of the resulting model better than SISA-FC, it entails more computational and storage costs than SISA-FC. Unlike SISA, these proposed methods do not provide exact unlearning guarantees, and their retain accuracy is lower than that of SISA. However, the training time and memory footprint of both methods is significantly lower than in SISA. A common drawback of both SISA-FC and SISA-A is that the content of the generic text corpus (learned during the pre-training phase) cannot be forgotten.

Zhang et al (2024a) introduce a method for combining parameter-efficient modules (PEMs) to enhance the capabilities of LLMs. The method utilizes linear arithmetic operations to compose different PEMs in ways that address specific needs like distribution generalization, multi-tasking, unlearning, and domain transfer. The unlearning aspect of this method employs negation operations to effectively remove or reduce undesired behaviors or knowledge from a model-akin to a detoxification process. This unlearning operation entails subtracting the parameters learned from a negative dataset from those learned from a standard dataset, resulting in a new unlearned PEM, thus enabling the model to “forget” these behaviors. For example, a PEM trained on toxic data can be negated to serve as a plug-in detoxifier, reversing its toxic effect.

4.3.2 Linear transformation

Belrose et al (2023) introduce LEACE, a closed-form method to prevent linear classifiers from detecting a concept, such as word part-of-speech, with minimal change of representation. This is achieved by applying a procedure called concept scrubbing, which erases all linear information about a target concept in every intermediate representation. LEACE sequentially fits an affine transformation to every intermediate layer to erase the target concept from its features while minimizing the distance from the original features. To fit LEACE parameters, samples from the respective model pre-training distribution are used. While this method can be used to improve fairness (*e.g.* preventing a classifier from using gender or race as deciding criteria) and interpretability (*e.g.* removing a concept to observe changes in model behavior), it has many limitations. It may degrade utility due to erasing features irrelevant to the concept and it requires caching hidden states during training, thereby leading to a substantial demand for memory. Also, its validation is limited to part-of-speech, and its practical applicability is uncertain.

Gender information in representations of biased LLMs can be divided into factual gender information and gender bias. Factual gender information encompasses grammatical or semantic properties indicating gender in English texts, such as explicit gendered pronouns like "he" or "she". Gender bias, on the other hand, refers to the model's tendency to associate certain words with specific genders, like "nurse" being more correlated with women than men. Limisiewicz and Mareček (2022) aim to mitigate the gender bias of a pre-trained LLM by manipulating contextual embedding. They apply an orthogonal transformation to separate lexical and syntactic information encoded in the model embedding. Then they filter out the bias subspace from the embedding space and keep the subspace encoding factual gender information. Although this method is efficient because it applies linear transformations on the contextual embedding only, there is no guarantee that all bias-related dimensions will be filtered. This is because the bias signal can be encoded non-linearly in LLMs and, even when the whole bias subspace is removed, the information can be recovered in the next layer of the model (Ravfogel et al 2020).

Leong et al (2023) propose a method for detoxifying LLMs by identifying and reversing toxification trajectories in the internal model activations. The method operates in two stages during inference: 1) generating internal activations for negative and positive prompts to determine the "toxification direction," and 2) linearly combining the target input's activations with the negated difference from Stage 1 to produce non-toxic outputs. Whereas this proposal is promising due to its simplicity and instance-level detoxification, it relies on carefully crafted prompts, which raises concerns about its ability to mitigate complex forms of bias and toxicity. Additionally, its effectiveness was primarily demonstrated on GPT-2, which raises questions about its generality across different models.

4.4 Input/output modification

Methods in this category treat the models as a black-box and only require access to the inputs and outputs of a model. This approach is specifically useful when the operator has no access to the model weights, *e.g.*, for products and services wrapping API-based models. However, in the context of the right to be forgotten, input/output modification does not yield any real privacy guarantee, as the models still retain the to-be-forgotten knowledge.

4.4.1 Input manipulation

In this approach, the data to be forgotten are deliberately altered or adjusted in a systematic way to facilitate the process of unlearning.

Gallegos et al (2024) propose a prompting-based bias mitigation method that leverages the zero-shot capabilities of LLMs to reduce biased stereotypical predictions and call it *zero-shot self-debiasing*. Two techniques, self-debiasing via explanation and self-debiasing via reprompting, are proposed. In the former, the model first explains invalid assumptions in the answer choices, whereby it identifies potential stereotyping. Then, it answers the question without stereotyping. In the second technique, the model first answers the question as per the baseline approach. Then, it is reprompted to remove bias from its initial answer.

The authors demonstrate the ability of their method to decrease stereotyping in question-answering over nine different social groups with a single prompt. Unlike traditional bias mitigations, this method does not require any additional training data, exemplar responses, fine-tuning, or auxiliary models, and thus it is a more efficient and practical solution for bias mitigation. However, the method is task and context-dependent. It is designed for multiple-choice questions, rather than for the more common open-ended question. Also, it uses manually created prompts, which limits its generalization to other biases. Future research could focus on detecting biases in free text and also exploring automated prompt generation to manage biases more effectively.

4.4.2 Information retrieval

The methods in this group aim to selectively extract or manipulate information from external knowledge to shape the unlearning trajectory of LLMs.

SERAC (Mitchell et al 2022) utilizes a memory-based model editing approach that treats an LLM as a black-box. It serves as a simple wrapper around the base model and consists of three main components: an explicit cache of edits, an auxiliary scope classifier, and a counterfactual model. When presented with a knowledge edit (which can be the unlearning request), the wrapped model predicts a new input in two steps. Firstly, the scope classifier assesses the likelihood of the new input falling within the scope of each cached edit. If deemed within scope, the edit with the highest probability is retrieved, and the counterfactual model provides a prediction based on both the new input and the retrieved edit. If the new input is considered out-of-scope for all edits, the base model's prediction is returned. Although SERAC was evaluated on editing question answering, fact-checking, and dialogue generation, it can also be used to unlearn undesirable behaviors of LLMs and replace them with desirable ones. This method is simple and easy to implement and requires no modifications to the original model. Also, it can edit multiple models with different architectures. However, it may be prone to retrieval errors, such as noise and harmful content, and knowledge conflict issues (Zhang et al 2024b). Further, it relies on an edit dataset for training and may require more computational and memory resources in some settings.

To correct model errors via user interactions without retraining, MemPrompt (Madaan et al 2022) pairs the model with a growing memory of recorded cases where the model misunderstood the user intent. The system maintains a memory of the feedback as a set of key-value pairs, where the key is a misunderstood input (*e.g.*, question), and the value is its correction. Given a new prompt, the system looks in the memory for a similar prompt to check if the model has made a mistake on a similar prompt earlier. If a match is found, the corresponding correction is appended to the prompt, and then the updated prompt is fed to the model. In this sense, this approach can be seen as an instance of prompt engineering (Liu et al 2023) which involves editing the prompts. The method was applied to lexical relations like antonyms, word scrambling such as anagrams, and ethics, where user feedback is used as the appropriate ethical consideration in natural language. While the method is efficient and gives the end-user more control over the unlearning process, it might struggle with the scalability of the memory or in maintaining utility as the volume of user interactions grows. Also, its effectiveness highly depends on the quality of user corrective feedback and on the success of matching the new input with its similar recorded one.

4.4.3 In-context learning

This approach exploits the in-context learning power of LLMs for unlearning.

To remove the impact of a specific training point on the model's output, ICUL (Pawelczyk et al 2023) constructs a specific context at inference time that makes the model classify it as if it had never seen the data point during training. The ICUL method involves three steps:

1. Label-flipping. Given a forgetting request, the label on the corresponding training point whose influence should be removed from the model is flipped, resulting in a new template.
2. Addition of correctly labeled training points. Excluding the to-be-forgotten point, s labeled example pairs are randomly sampled and added to the template from Step 1.
3. Prediction. The query input is added to the template, forming the final prompt, and the model predicts the next token using a temperature of 0.

The label-flipping operation in Step 1 aims to remove the influence of a specific training point on the model outcome. Step 2 aims to reduce the effect of the label flipping, with the number of points s allowing for a trade-off between efficiency and utility. The ICUL method was empirically evaluated on three classification datasets using a test called LiRA-Forget, which was also introduced to empirically measure unlearning effectiveness. The results demonstrate that ICUL can effectively eliminate the influence of training points on the model's output, occasionally outperforming white-box methods that necessitate direct access to the LLM parameters and are more computationally demanding. While this method is efficient and preserves the utility of the model, its effectiveness depends on the model's capacity for in-context learning. It is worth noting that its efficacy was only tested with text classification tasks where the label-flipping process is applicable. Its effectiveness on other tasks remains to be determined.

Table 1 summarizes and compares the LLM unlearning methods surveyed in this paper under the types and requirements of digital forgetting discussed in Sect. 2.

5 Evaluation of unlearning in LLMs

In this section, we study the evaluation of unlearning in LLMs, including the datasets, models, and metrics employed by the surveyed works. Our analysis of metrics focuses on the main aspects discussed in Sect. 2.3: (i) whether the model has effectively forgotten the target knowledge, (ii) whether the model retained the rest of its capabilities, and (iii) the computational cost of the forgetting process. Hereafter, these measures will be referred to as forgetting, retaining, and runtime, respectively.

5.1 Datasets

A proper assessment of forgetting generally requires three different datasets, which we refer to as *forgetting training set*, *forgetting test set*, and *retaining test set*. The two forgetting sets are the unlearning counterparts of the training and test sets employed during training: the forgetting training set comprises samples representing the knowledge that the model should forget, whereas the forgetting test set facilitates the measurement of the forgetting generalization. The retaining test set is typically disjoint from the forgetting sets and is used to assess the utility of the unlearned model.

Table 1 Comparison between LLM unlearning methods. L, M and H denote low, medium and high, respectively. Sgnf denotes significant

Method	Forgetting	Target	Application						Retaining	Runtime
			Privacy	Copyright	Robustness	Toxicity	Fairness	Capabilities		
Global weight modification										
SISA (Bourtoule et al 2021)	Exact	Item	•	•	•	•	–	–	M-L	Sgnf
LOO (Liu and Kalinli 2023)	Approx	Item	•	•	•	•	–	–	H-M	H
GA (Jang et al 2022)	Approx	Item	•	•	–	•	–	–	M-L	M
SeUL (Wang et al 2024)	Approx	Item	•	•	–	•	–	–	M	H
Yao et al (2023)	Approx	Item	•	•	•	•	–	–	M	H
KGA (Wang et al 2023)	Approx	Item	•	•	•	•	•	•	M-L	H
Eldan and Russinovich (2023)	Approx	Concept	•	•	–	•	–	–	H-M	H
Quark (Lu et al 2022)	Approx	General	•	•	•	•	•	•	H	Sgnf
Local weight modification										
PCGU (Yu et al 2023)	Approx	Concept	–	–	–	–	•	–	H	M
Task vector (Ilharco et al 2022)	Approx	Task	–	–	–	–	–	•	H-M	H
F-Learning (Ni et al 2023)	Approx	Task	–	–	–	–	–	•	H-M	H
DEPN (Wu et al 2023)	Approx	Item	•	–	–	–	–	–	M-L	M
Pochinkov and Schoots (2023)	Approx	Task	–	–	–	–	–	•	H	M
Architecture modification										
EUL (Chen and Yang 2023)	Weak/No	Item	•	•	•	•	–	–	H-M	M
SISA-FC (Kumar et al 2022)	Weak/No	Item	•	•	•	•	–	–	L	L
SISA-A (Kumar et al 2022)	Weak/No	Item	•	•	•	•	–	–	M	M
Zhang et al (2024a)	Weak/No	Task	–	–	–	•	–	–	M	M
LEACE (Belrose et al 2023)	Weak/No	Concept	–	–	–	–	•	–	H-M	M
Limisiewicz and Mareček (2022)	Weak/No	Concept	–	–	–	–	•	–	H-M	L
Leong et al (2023)	Weak/No	Item	–	–	–	•	–	–	H-M	L
Input/output modification										
Gallegos et al (2024)	Weak/No	Concept	–	–	–	–	•	–	H	L
SERAC (Mitchell et al 2022)	Weak/No	General	•	•	•	•	•	•	H	M
MemPrompt (Madaan et al 2022)	Weak/No	General	•	•	•	•	•	•	M	L
ICUL (Pawelczyk et al 2023)	Weak/No	General	•	•	•	•	•	•	H	L

Table 2 lists the forgetting datasets used by the surveyed methods. The “Forgetting target” column specifies the forgetting request type (as defined in Sect. 2.2) and a description of the undesired knowledge that has to be unlearned. Note that some authors select combinations of undesired knowledge and forgetting request types that do not align with real-world scenarios, although their choices might be justified for experimental purposes. On the one hand, Wu et al (2023) and Borkar (2023) consider PII as *items* to be forgotten. However, sensitive information should be treated as a *concept* to be forgotten, to preclude the model from providing any details compromising privacy. For example, for an individual born in Paris, even if the model does not generate the city name verbatim, it can disclose that the birthplace is ’The city of light’, the sobriquet of Paris. On the other hand, Eldan and Russinovich (2023) experiment with forgetting a copyrighted corpus (Harry Potter books) as a concept. This may be excessive to avoid copyright laws, for which it is sufficient not to generate copies of the text. Forgetting copyrighted texts as *items* should be enough for these cases.

Table 3 lists datasets used to assess the retaining of models. The “Retaining target” column specifies the preserved capability under evaluation. Note that the zsRE (Levy et al 2017) dataset and those subsequently listed were also employed for forgetting purposes by the same authors (see Table 2). This is because the objective was to selectively forget a subset of samples, such as specific generations (*i.e.*, private information in Enron emails (Klimt and Yang 2004)), a concept (*i.e.*, gender bias in Bias in Bios (De-Arteaga et al 2019)) or a task (*e.g.*, Python programming in CodeParrot GitHub Code (Tunstall et al 2022)), while retaining the rest of dataset-related skills. Several researchers choose widely recognized benchmarks for LLMs (Eldan and Russinovich 2023; Jang et al 2022; Yao et al 2023; Pawelczyk et al 2023; Chen and Yang 2023) to evaluate the preservation of overall capabilities. In contrast, a limited number of authors employ datasets with tasks that are closely aligned with the knowledge intended for forgetting (Limisiewicz and Mareček 2022; Belrose et al 2023; Pochinkov and Schoots 2023; Ni et al 2023). As it will be detailed in Sect. 5.4, this strategy is taken because tasks that are similar but peripheral to the focus of the forgetting process are expected to be the most affected by it.

5.2 Models

Table 4 reports the LLMs used by each of the surveyed methods. Models are sorted by their number of parameters, that differ by as much as 3 orders of magnitude (*i.e.*, from 11 million to 30 billion parameters). LLMs are often released in various sizes to accommodate different use cases. The table enumerates the specific sizes used, separated by commas. In cases where multiple authors selected the same model but in different sizes, a row without the model name is used. As suggested by Carlini et al (2021), larger models tend to memorize more, and this makes them more challenging and interesting for evaluation. Notably, 8 works opted for models with over 1 billion parameters.

5.3 Forgetting evaluation

Most surveyed methods only offer approximate unlearning. Therefore, the forgetting success must be evaluated empirically. This empirical evaluation is conducted by measuring the model’s capabilities before and after the unlearning process via some metrics. Namely,

Table 2 Datasets used for forgetting

Dataset name	Forgetting target	Paper/s
PKU-SafeRLHF (Ji et al 2023)	Concept: Harmful generations	Yao et al (2023)
HaluEval (Li et al 2023a)	Concept: Hallucinations	Yao et al (2023)
RealToxicityPrompts (Gehman et al 2020)	Concept: Toxicity	Lu et al (2022); Ilharco et al (2022)
Civil Comments (Borkan et al 2019)	Concept: Toxicity	Ilharco et al (2022)
Harry Potter Universe (Eldan and Russinovich 2023)	Concept: Related generations	Eldan and Russinovich (2023)
WinoBias (Zhao et al 2018)	Concept: Gender bias	Limisiewicz and Mareček (2022)
WinoMT (Stanovsky et al 2019)	Concept: Gender bias	Limisiewicz and Mareček (2022)
CrowS Pairs (Nangia et al 2020)	Concept: Gender bias	Yu et al (2023)
Winogender Schemas (Rudinger et al 2018)	Concept: Gender bias	Yu et al (2023)
Bias in Bios (De-Arteaga et al 2019)	Concept: Gender bias	Belrose et al (2023)
StereoSet (Nadeem et al 2021)	Concept: Stereotype bias	Yu et al (2023)
English Universal Dependencies (Nivre et al 2020)	Concept: Part-of-Speech	Belrose et al (2023)
RedPajama (Computer 2023)	Concept: Part-of-Speech	Belrose et al (2023)
Training Data Extraction Challenge [4]	Item: Specific generations	Jang et al (2022)
Pile (Gao et al 2021)	Item: Samples & Concept: PoS	Jang et al (2022); Belrose et al (2023); Pochinkov and Schoots (2023)
Harry Potter and the Sorcerer's Stone (Rowling 2000)	Item: Copyright corpus	Yao et al (2023)
SST2 (Socher et al 2013)	Item: Samples	Pawelczyk et al (2023)
Amazon polarity (Zhang et al 2015)	Item: Samples	Pawelczyk et al (2023)
Yelp polarity (Zhang et al 2015)	Item: Samples	Pawelczyk et al (2023)
IMDB (Maas et al 2011)	Item: Samples	Chen and Yang (2023)
SAMSum (Gliwa et al 2019)	Item: Samples	Chen and Yang (2023)
LEDGAR (Tuggener et al 2020)	Item: Samples	Wang et al (2023)
PersonaChat (Zhang et al 2018)	Item: Samples	Wang et al (2023)
IWSLT14 (Cettolo et al 2014)	Item: Samples	Wang et al (2023)
Enron emails (Klimt and Yang 2004)	Item: Private information	Wu et al (2023); Borkar (2023)
zsRE (Levy et al 2017)	Item: Old facts	Ni et al (2023)
CounterFact (Meng et al 2022)	Item: Old facts	Ni et al (2023)
CodeParrot GitHub Code (Tunstall et al 2022)	Task: Programming	Pochinkov and Schoots (2023)

[4] https://github.com/google-research/lm-extraction-benchmark

the model's capabilities related to the behavior the model is supposed to unlearn should decrease. We can further evaluate whether the unlearning process can generalize through the forgetting training and test sets. Considering the capabilities of LLMs for memorizing text (Carlini et al 2021), it is possible that the forgetting process only works for samples in the forgetting training set, and even a slight paraphrasing could be sufficient to obtain undesired predictions.

Table 3 Datasets only used for retaining

Dataset name	Retaining target	Paper/s
Wizard of Wikipedia (Dinan et al 2019)	Dialogue	Jang et al (2022)
Empathetic Dialogues (Rashkin et al 2019)	Dialogue	Jang et al (2022)
Blended Skill Talk (Smith et al 2020)	Dialogue	Jang et al (2022)
Wizard of Internet (Komeili et al 2022)	Dialogue	Jang et al (2022)
piqa (Bisk et al 2020)	Q&A	Eldan and Russinovich (2023); Jang et al (2022)
COPA (Gordon et al 2012)	Q&A	Jang et al (2022)
ARC (Clark et al 2018)	Q&A	Jang et al (2022)
MathQA (Amini et al 2019)	Q&A	Jang et al (2022)
PubmedQA (Jin et al 2019)	Q&A	Jang et al (2022)
TruthfulQA (Lin et al 2022)	Q&A	Yao et al (2023)
GAP Coreference (Webster et al 2018)	Coreference	Limisiewicz and Mareček (2022)
English Web Treebank (Silveira et al 2014)	Dependency	Limisiewicz and Mareček (2022)
WinoGrande (Sakaguchi et al 2020)	Completion	Eldan and Russinovich (2023); Jang et al (2022)
HellaSwag (Zellers et al 2019)	Completion	Eldan and Russinovich (2023); Jang et al (2022)
Lambada (Paperno et al 2016)	Completion	Jang et al (2022)
zsRE (Levy et al 2017)	Completion	Ni et al (2023)
CounterFact (Meng et al 2022)	Completion	Ni et al (2023)
Pile (Gao et al 2021)	Samples subset	Jang et al (2022); Pochinkov and Schoots (2023)
SST2 (Socher et al 2013)	Samples subset	Pawelczyk et al (2023)
Amazon polarity (Zhang et al 2015)	Samples subset	Pawelczyk et al (2023)
Yelp polarity (Zhang et al 2015)	Samples subset	Pawelczyk et al (2023)
IMDB (Maas et al 2011)	Samples subset	Chen and Yang (2023)
SAMSum (Gliwa et al 2019)	Samples subset	Chen and Yang (2023)
LEDGAR (Tuggener et al 2020)	Samples subset	Wang et al (2023)
PersonaChat (Zhang et al 2018)	Samples subset	Wang et al (2023)
IWSLT14 (Cettolo et al 2014)	Samples subset	Wang et al (2023)
Enron emails (Klimt and Yang 2004)	Generation	Wu et al (2023); Borkar (2023)
CodeParrot GitHub Code (Tunstall et al 2022)	Generation	Pochinkov and Schoots (2023)
WikiText-103 (Merity et al 2016)	Generation	Ilharco et al (2022)
Bias in Bios (De-Arteaga et al 2019)	Classification	Belrose et al (2023)

In the following, we examine the metrics employed by the surveyed works to measure the attained level of forgetting:

- **Dataset-specific metrics**: Most papers (Eldan and Russinovich 2023; Jang et al 2022; Yao et al 2023; Pawelczyk et al 2023; Chen and Yang 2023; Ni et al 2023; Pochinkov and Schoots 2023) leverage the predefined performance metric of the forgetting dataset, that is, the accuracy in a classification or completion dataset.
- **Toxicity**: Papers seeking to mitigate the toxicity of generated texts (Ilharco et al 2022; Lu et al 2022) sometimes rely on off-the-shelf metrics, such as Perspective API[4] or Tox-

[4] https://github.com/conversationai/perspectiveapi

Table 4 LLMs used for forgetting

Model name	Number of parameters	Paper/s
ALBERT (Lan et al 2020)	11 M	Yu et al (2023)
DistilBERT (Sanh et al 2019)	67 M	Wang et al (2023)
BERT (Devlin et al 2019)	110 M	Wu et al (2023); Yu et al (2023); Limisiewicz and Mareček (2022); Belrose et al (2023)
ELECTRA (Clark et al 2020)	110 M	Limisiewicz and Mareček (2022)
RoBERTa (Liu et al 2019)	355 M	Pochinkov and Schoots (2023)
	125 M	Yu et al (2023)
GPT-2 (Radford et al 2019)	117 M, 355 M, 774 M	Ilharco et al (2022)
	355 M, 774 M	Lu et al (2022)
Bloom (Scao et al 2022)	560 M, 1.1B	Pawelczyk et al (2023)
Phi (Li et al 2023b)	1.5B	Eldan and Russinovich (2023)
GPT-Neo (Black et al 2022)	125 M, 1.3B, 2.7B	Jang et al (2022)
T5 (Raffel et al 2020)	223 M, 3B	Chen and Yang (2023)
OPT (Zhang et al 2022)	125 M, 1.3B, 6.7B	Pochinkov and Schoots (2023)
	125 M, 1.3B, 2.7B	Jang et al (2022)
	1.3B, 2.7B	Yao et al (2023)
Galactica (Taylor et al 2022)	125 M, 1.3B, 6.7B	Pochinkov and Schoots (2023)
Llama-2 (Touvron et al 2023b)	7B	Yao et al (2023); Ni et al (2023)
Pythia (Biderman et al 2023)	160 M, 1.4B, 6.9B, 12B	Belrose et al (2023)
	160 M, 1.4B, 6.9B	Pochinkov and Schoots (2023)
Llama (Touvron et al 2023a)	7B, 13B, 30B	Belrose et al (2023)
	7B	Eldan and Russinovich (2023)

icity.[5] These metrics involve training an LLM on a toxic dataset to predict the toxicity score of input text.

- **Bias**: Works addressing bias in LLMs (*e.g.*, gender, race or stereotype) (Belrose et al 2023; Limisiewicz and Mareček 2022; Yu et al 2023) commonly rely on the bias-sensitive prediction probability. For example, for pronoun prediction of a person with a known occupation (*e.g.*, doctor), probabilities for both pronouns are expected to be 50%. Under the premise of profession-induced gender bias, De-Arteaga et al (2019) introduced the TPR-GAP metric, which assesses the difference (GAP) in the true positive rate (TPR) between the two genders for each occupation. On the same basis, Limisiewicz and Mareček (2022) presented the relative gender preference (RGP) metric, which can be roughly defined as the difference in gender probabilities for texts with and without the profession name.
- **Generation**: For undesired knowledge with no specific metrics, the evaluation often focuses on how easily the model can generate that undesired knowledge. The evaluation will depend on whether the forgetting process aims to avoid generating text that exactly or approximately matches that of the forgetting request:

  – *Exact generation*: The goal is to prevent the model from generating verbatim copies of the text to be forgotten. This is common in scenarios involving copyright. To that end, the perplexity metric is often employed (Jang et al 2022; Yao et al 2023; Wu et al 2023; Pochinkov and Schoots 2023). This approach is sometimes followed when

[5] https://github.com/unitaryai/Toxicity

the objective is to avoid the generation of PII. Jang et al (2022) propose the extraction likelihood metric, which considers overlaps of n-grams, and the memorization accuracy metric, for matching identical token sequences. They define unlearning as complete when both metrics are lower than a threshold for the forgetting test set. Wu et al (2023) adapt their metrics to the type of PII to be forgotten, such as exposure for phone numbers and mean reciprocal rank for names. Note that most of these evaluations do not use a forgetting test set, but focus on *exact* replicates. This approach may not be enough to deal with privacy issues, since we should prevent the generation of the targeted sensitive concept, not only the concrete way it is expressed (unlike copyright).

  – *Similar generation*: In this scenario, we consider any type of generations which suggest that the model has been trained with undesired knowledge. For instance, Wang et al (2023) measure output similarity with the text to be forgotten by using the Jensen-Shannon Divergence (JSD), the Language model Probability Distance (LPD), and the Proportion of instances with Decreased Language model Probability (PDLP), while Yao et al (2023) leverage the BiLingual Evaluation Understudy (BLEU) score. Eldan and Russinovich (2023), who aimed to forget the Harry Potter universe corpora, opted for a more fine-grained evaluation, using a forgetting test set curated to reflect knowledge of the Harry Potter books.

- **Membership Inference Attack (MIA)**: A few authors (Wang et al 2023; Chen and Yang 2023; Pawelczyk et al 2023) use the success of a MIA as a metric. This type of attack aims to determine whether a piece of knowledge was used during the (pre-)training of the model. See Shokri et al (2017); Yeom et al (2018) for a primer on MIAs, Carlini et al (2021); Nasr et al (2023) for examples of black-box MIA on LLMs, and Patil et al (2023) for an example of a white-box MIA on LLMs.

Among the surveyed works, we have found two main approaches for the evaluation of forgetting. The first, a more realistic scenario contemplated by Jang et al (2022); Yao et al (2023); Yu et al (2023); Limisiewicz and Mareček (2022), involves applying forgetting techniques to a standard, publicly available, pre-trained LLM with pre-existing biases from its training data. This approach has some constraints, notably that the undesired knowledge must be well-represented in the pre-training dataset and that an external dataset may be needed if the specific undesired data are not identifiable enough. Due to these limitations, some researchers, such as Eldan and Russinovich (2023); Chen and Yang (2023); Borkar (2023); Wu et al (2023); Wang et al (2023) prefer a second, more controllable but less realistic scenario, where forgetting is applied to an LLM fine-tuned with a dataset that includes the undesired knowledge. This allows for better control over the process, with the original, unaltered LLM serving as a baseline for comparison. However, this scenario risks unrealistic results due to the potential overfitting on the undesired knowledge.

5.4 Retaining evaluation

The assessment of model retaining often involves evaluating the utility difference of the LLM before and after the forgetting process on tasks different from the undesired knowledge. The smaller the utility difference, the better the retention. Ideally, the retention evaluation should encompass all possible knowledge except that to be forgotten. Nonetheless, this is unfeasible in most cases, given the wide range of capabilities that LLMs can learn. Instead, researchers define the forgetting test set by selecting the subset of the most relevant tasks for which utility should be retained.

In the following, we list the most prominent strategies used to measure the retained utility:

- **General benchmarks**. Eldan and Russinovich (2023), Wang et al (2023), and Jang et al (2022) leverage well-established LLM benchmarks, such as those used for comparing LLMs (Sakaguchi et al 2020; Zellers et al 2019; Paperno et al 2016; Bisk et al 2020). These benchmarks assess the model's capabilities in NLU, reasoning, and human-like generation. The performance of the original and resulting models in these general tasks is compared to assess retaining.
- **Related but different**. A reasonable assumption is that knowledge conceptually close to that to be forgotten will be the most affected by the unlearning process. Therefore, the performance of the models in these domains may be an appropriate indicator of whether the model is retaining the remaining capabilities or unlearning is affecting more than just the undesired knowledge. Following this principle, several authors evaluate retaining in domains related but different to the undesired knowledge (Yao et al 2023; Lu et al 2022; Limisiewicz and Mareček 2022; Belrose et al 2023; Chen and Yang 2023; Wang et al 2023; Pawelczyk et al 2023; Wu et al 2023; Ni et al 2023; Pochinkov and Schoots 2023). The specific strategies employed differ in terms of closeness and/or relatedness to the forgetting target:

  - *Related dataset*. Lu et al (2022); Yao et al (2023); Limisiewicz and Mareček (2022); Ilharco et al (2022) choose datasets different from but related to those used for unlearning. For instance, Lu et al (2022) measured toxicity, fluency and diversity in the WritingPrompts dataset (Fan et al 2018) to measure the retaining of the model's writing capabilities when forgetting toxic generations from the RealToxicPrompts dataset (Gehman et al 2020). Similarly, Yao et al (2023) leveraged BLEURT (Sellam et al 2020) and the *deberta-v3-large-v2* reward model[6] to compare the generations for TruthfulQA (Lin et al 2022) questions by the original LLM with those resulting from unlearning harmful Q&A pairs of PKU-SafeRLHF (Ji et al 2023).
  - *Same dataset but different samples*. Chen and Yang (2023); Wang et al (2023); Pawelczyk et al (2023); Ni et al (2023); Pochinkov and Schoots (2023) evaluate retaining on the remaining samples of the same dataset used for unlearning. For example, Pochinkov and Schoots (2023) forget Python code samples from Tunstall et al (2022) while aiming to retain the remaining code samples.
  - *Same dataset but different task*. The forgetting dataset is used to measure the model's capabilities in a task disjoint of that to be forgotten (Belrose et al 2023; Wu et al 2023). For instance, the Bias in Bios dataset (De-Arteaga et al 2019) is employed by Belrose et al (2023) for both forgetting gender bias and measuring retaining in the job prediction task. Another example is provided by Wu et al (2023), who aim to forget PII from the Enron dataset (Klimt and Yang 2004), and measure retaining as the generation quality for the rest of the corpus.

[6] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2

The utility degradation observed in these studies usually falls within the range of 1–5%, although it can occasionally reach up to 20%. Jang et al (2022) and Pawelczyk et al (2023) show that larger models tend to exhibit better utility retention.

5.5 Runtime evaluation

The runtime of unlearning methods should always be significantly lower than the runtime of retraining the models from scratch, which is the baseline for unlearning. However, unlearning is rarely a fast procedure, since it usually involves costly training-related steps, such as fine-tuning.

Several of the surveyed works explicitly report the runtime of their methods. Reported values range from minutes (Wang et al 2023; Yao et al 2023; Limisiewicz and Mareček 2022; Jang et al 2022; Yu et al 2023) to hours (Chen and Yang 2023) or even longer than a day (Lu et al 2022). However, directly comparing the runtime reported by different works might be unfair, due to the different hardware configurations, models, and/or undesired knowledge.

There are some workarounds to compensate for hardware discrepancies, although they are not universally applicable. First, the runtime can be normalized to the most common GPU, adjusting runtime from other GPUs based on their relative performance in deep learning, as assessed through specific benchmarks. However, the substantial memory requirements of LLMs often require multi-GPU configurations with the corresponding interconnections. The latency of these interconnections also influences runtime, but this is rarely reported in the literature. Second, for those methods whose time cost is predominantly determined by fine-tuning, the number of epochs can be used as a more general indicator of cost. However, this information is only reported by a subset of works (Eldan and Russinovich 2023; Chen and Yang 2023; Jang et al 2022; Yao et al 2023; Yu et al 2023; Wu et al 2023; Ni et al 2023; Ilharco et al 2022) and it can only be used when comparing identical models, data and forgetting tasks, which is rarely the case.

6 Challenges and recommendations

This section outlines several challenges of machine unlearning we have identified when considering the technical and legal requirements of digital forgetting and the surveyed works. Our goal is to guide researchers and practitioners in: i) the identification and application of unlearning techniques that are suitable to their needs; and ii) the evaluation of the effectiveness of those techniques.

6.1 Managing forgetting request types

A first challenge for practitioners is how to process forgetting or unlearning requests. We start with a description of general forgetting requests as implemented to comply with the right of erasure described in the GDPR (European Parliament and Council of the European Union, 2016).

The right to erasure in Article 17 of the GDPR states that: “The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay [...]". The regulation does not specify any form or information the data subject has to provide to the data controller to exercise their right. Search engines, such as Google Search, provide specific forms to exercise the right to be forgotten. In their forms, Google requests the data subjects to identify, and then provide a list of URLs that contain personal information about them, which specific search queries point to the documents of interest, and the motivation for the erasure.[7] Therefore, the implementation of the right to be forgotten revolves around removing *documents* that are present on the internet and contain some personal information about the data subject from search results. This erasure does not imply *all* information about a data subject has to be deleted from search results, but only that information requested by the data subject.

How these requests translate to large generative models, and LLMs in particular, and to requests other than those concerned with data privacy is a subject to be further studied, but we should expect requests that include the problematic outputs of LLMs along with the inputs (or prompts) that triggered them. Given that information, the model owner will be able to determine the nature of the problematic outputs and decide whether the issue is related to privacy or copyright, to the generation of non-factual or incorrect information, or to the generation of outputs that are not aligned with human values, such as biased, toxic, or harmful information.

The unlearning mechanisms surveyed in Sect. 4, along with other forgetting mechanisms mentioned in Sect. 3 apply to different unlearning targets, with different guarantees, evaluation methods, and costs. Thus, the choice of a forgetting or unlearning mechanism will depend on the type of knowledge to be forgotten and on the balance of three, often conflicting, properties, namely, the unlearning guarantees provided by the mechanism, the utility of the resulting model, and the execution time of the solution.

We next summarize our findings on the available unlearning mechanisms, describing the available options for each of the possible unlearning requests.

Forgetting procedures aimed at removing private information are covered by many of the works we have surveyed. One of the reasons for that is that privacy has been a topic of research for several years, and definitions of privacy originally thought for databases have carried over into the machine learning field. Not surprisingly, the unlearning guarantees defined by Nguyen et al (2022) and Xu et al (2023) are heavily inspired by differential privacy, and privacy attacks, such as membership inference attacks or reconstruction attacks, have been used to measure the risk of privacy leaks and to evaluate the effectiveness of unlearning. This has resulted in a very rich literature on unlearning mechanisms dealing with private information, even in the context of LLMs. Therefore, practitioners can choose from a wide range of methods, which include methods that provide exact guarantees but at high computing cost, such as SISA (whose set-up cost is equivalent to that of retraining from scratch, but which makes the processing of unlearning requests faster) as well as methods offering no guarantees but fast execution, such as ICUL. Even when no *ex ante* guarantees exist, evaluation of unlearning for privacy (which amounts to evaluating whether the model knows a particular information) has been studied for years, and techniques such as MIAs can be used for empirical evaluation (Carlini et al 2021; Nasr et al 2023). One of the main challenges of unlearning private information is that most methods focus on unlearning *items* or data points from the models. Whereas that might be enough to remove PII such as identi-

[7] Google's Right to be Forgotten form – https://reportcontent.google.com/forms/rtbf

fiers or specific sensitive information, it might not suffice for quasi-identifiers (attributes of individuals that are not direct identifiers but that im combination can uniquely identify an individual). Unlearning approaches that deal with this issue should not be devoted to deleting specific information about individuals, but to removing any possible correlations between someone's identity and all the set of attributes that could describe that person.

Addressing copyright issues through unlearning is quite similar to the case of privacy. All but one of the works we have surveyed that apply to privacy also apply to copyright issues, as do guarantees, costs, and evaluation mechanisms and criteria. However, we find a distinction between forgetting or unlearning copyrighted material –needed when the rightholders forbid any processing of their protected work by the developers of an LLM–, where the approaches described above would apply, and ensuring that generative models do not output copyrighted material, which could constitute plagiarism. In the latter case, an unlearning procedure would just need to ensure that no verbatim fragments of a protected work are generated by the LLM. There is, however, a right to quote works for the purposes of commentary or criticism. Therefore, additional measures should be taken to make LLMs attribute quotes accurately. This is an open problem that requires more study.

When forgetting requests refer to model robustness, where we include the removal of factually incorrect information, fake news, or outdated information, we face a similar situation to that of privacy or copyright protection. Note that, according to the works we have surveyed, methods that target *items* apply seamlessly to several of the issues we want to correct using unlearning. The reason is that an item is a piece of data that can be as small as a token or as big as a whole document. As above, choosing which mechanism to use will depend on the desired guarantees and runtime.

Finally, we cover alignment issues, which include discrimination, the generation of toxic, harmful, or hateful outputs, and the unlearning of unwanted capabilities. Regarding methods that focus on fairness or the mitigation of biases and discrimination, our survey has revealed that most studies focus on gender discrimination related to pronouns (e.g., "doctor" is more frequently associated with "him" while "nurse" is more frequently associated with "her"). While addressing this issue is crucial, we believe it represents a limited perspective on bias, which in fact manifests across many more dimensions. Thus, we observe that more research should be conducted in this domain, covering different aspects of discrimination and bias. Addressing all potential biases in text documents could significantly impact the models' retaining performance. Also, the lack of a clear definition or methodology for approaching bias complicates the evaluation of bias mitigation efforts. Each type of bias –such as gender, ethnicity, or race– may require unique evaluation datasets and methods. Furthermore, most methods designed to correct biases offer little to no concrete guarantees, requiring practitioners to continuously monitor and assess any potential biases present in their models. It may be beneficial to discuss the need for standardized frameworks or benchmarks that could provide a more systematic approach to evaluate and address biases in AI models. An even less concrete type of requests are those related to the generation of toxic, harmful, hateful, or hallucinated content. These phenomena are difficult to define in many cases because of cultural differences, the use of humor and sarcasm, the importance of context, etc. Methods that deal with these issues are often based on further fine-tuning and input and output manipulation, and therefore only provide at most weak guarantees of unlearning. For toxic and hateful content, evaluation can be carried out leveraging other LLMs such as HateBERT.

We have observed that methods dealing with alignment issues (with the exception of the generation of toxic content, which in some cases can be tackled with the same mechanisms for privacy and copyright issues) have garnered less attention in the literature. As discussed above, a possible reason is that defining what is and what is not undesirable knowledge is difficult in the alignment scenario. Most methods, therefore, offer at most approximate guarantees; even in this case, the empirical evaluation of the unlearning performance is not well established.

6.2 Guarantees of forgetting

We now discuss the limitations of *ex ante* guarantees of forgetting.

The forgetting guarantees presented in Sect. 2.3 were mainly devised for general neural network classifiers. Although they can still be used for LLMs, some factors limit their applicability. On the one hand, the guarantees refer to the parameters of the resulting models after training or forgetting. This means that, for an *ex ante* guarantee of forgetting, a clear approximation of the influence of every training sample on the model weights is known or can be computed. Whereas this may be feasible for small or simpler models, the complexity of LLMs can make such analyses difficult if not impossible.

On the other hand, although unlearning an image or a record is relatively straightforward in typical machine unlearning, the complexity increases when dealing with text data in LLMs. If the goal is to unlearn a text sequence containing sensitive information, not only is identifying explicit sensitive data difficult, but identifying implicit sensitive data (data that may allow sensitive inferences) is even more challenging. This challenge also applies when unlearning copyrighted documents, especially in non-fiction texts, where identifying the unique tokens associated with these documents becomes tedious. When the aim is to eliminate model bias or hallucination, the complexity increases further. Bias is often widespread and difficult to detect in training text data, as it appears in scattered patterns across many examples in both explicit and implicit forms. Thus, unlearners often resort to identifying biased concepts within internal model representations and addressing them at this level. Similarly, addressing hallucinations poses a significant challenge, as they often stem from multiple sources, making identification a nontrivial task.

6.3 Evaluation

Regarding the empirical evaluation of unlearning and retaining capabilities, we have found that the surveyed works often use specific datasets, and this may lower the applicability of many of these evaluations to the general case. To ensure the methods are not dataset-specific, diverse datasets for the same target task must be used.

Unlearning in LLMs is more challenging than in traditional classification models due to the vast output space of language models (their outputs are not just a class label), higher efficiency requirements, and limited access to training data. This makes evaluations difficult. Currently, no *de facto* standard datasets exist to explicitly evaluate unlearning methods. Researchers typically assess these methods across diverse datasets based on individual considerations.

Another challenge is that all methods are evaluated on English language datasets and there is no evidence about their performance with other languages. More datasets on different unlearning applications are required to assess how well unlearning methods generalize across different languages.

Furthermore, even if unlearning is motivated by concerns about private information leakage, the datasets used in the literature rarely contain sensitive information. To better validate the effectiveness of unlearning for sensitive information, future research should use simulated datasets with representative distributions of sensitive information.

6.4 Balancing forgetting requirements

In order for practitioners and operators of LLM-based services to comply with forgetting requests, whatever their type and target, they can follow a tiered approach that balances the runtime of the solutions (in order to comply with requests in a timely manner) against the forgetting guarantees and the retain performance.

Input/output modification approaches, described in Sect. 4.4, along with output post-processing methods, can be deployed almost instantly to steer the model behavior towards a more desirable one without impacting the retain performance in any significant way. However, these approaches can only provide weak unlearning guarantees, which in some cases can be reverted by users using carefully crafted prompts. Although these approaches can serve as a fast hotfix to LLMs, they are only recommended as a short-term approach while other, more robust but more costly, approaches are applied in parallel.

Unlearning methods based on local modifications or fine-tuning offer a better balance between speed, unlearning guarantees, and utility of the unlearned models. All of the surveyed methods in these categories offer approximate guarantees, which in most cases should be sufficient to comply with forgetting requests, especially those related to privacy and copyright issues (dealing with biases and the generation of harmful content is a more complex issue). These methods can be used to consolidate changes made with the previous approaches without the need to retrain the models.

A more definitive approach to complying with forgetting requests is to retrain the models from scratch, applying any necessary data pre-processing or privacy-preserving training techniques. Mechanisms such as SISA can also be applied in order to prepare for future unlearning requests. All these approaches provide exact guarantees and a good utility of the resulting models, but they require significant resources.

7 Conclusions

We have surveyed the state of the art in digital forgetting and machine unlearning in LLMs from several points of view.

First, we have reviewed and characterized the motivation, the types, the requirements and the main approaches to digital forgetting in LLMs. Next, we have conducted the most comprehensive and up-to-date survey on unlearning methods for LLMs in the literature, encompassing 22 works, most of them published in 2023 and 2024. Unlearning methods have been organized according to a fine-grained taxonomy, which classifies them according to the type of modification they perform on the model and their fundamental working principle. We have also surveyed the way the proposed methods are evaluated for forgetting, retaining and runtime; we have identified 28 datasets used for evaluation of forgetting, 31 datasets used for evaluation of retaining, and 15 LLMs used by the surveyed methods (plus size variations).

Finally, we have identified the most outstanding challenges of machine unlearning in LLMs, and we have proposed recommendations for practitioners on how to choose specific methods depending on their needs. We believe the scientific community and the professionals developing, managing, and deploying models can benefit from our recommendations on how to apply machine unlearning methods to satisfy their needs, possibly imposed by the existing regulations.

Acknowledgements This work has been funded by Huawei Technologies Finland Research Center. Partial support to this work has been received from the European Commission (project H2020-871042 "SoBigData++"), the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and to D. Sánchez, and grant 2021SGR-00115), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" under grant PID2021-123637NB-I00 "CURLING", and the EU's NextGenerationEU/PRTR via INCIBE (project "HERMES" and INCIBE-URV cybersecurity chair).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abadi M, Chu A, Goodfellow I, et al (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 308–318

Amini A, Gabriel S, Lin S, et al (2019) MathQA: Towards interpretable math word problem solving with operation-based formalisms. In: Burstein J, Doran C, Solorio T (eds) 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). Association for Computational Linguistics, Minneapolis, Minnesota, pp 2357–2367, https://doi.org/10.18653/v1/N19-1245, https://aclanthology.org/N19-1245

Belrose N, Schneider-Joseph D, Ravfogel S, et al (2023) Leace: Perfect linear concept erasure in closed form. Preprint at arXiv:2306.03819

Biderman S, Schoelkopf H, Anthony QG, et al (2023) Pythia: A suite for analyzing large language models across training and scaling. In: International Conference on Machine Learning, PMLR, pp 2397–2430

Bisk Y, Zellers R, Bras RL, et al (2020) PIQA: reasoning about physical commonsense in natural language. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, pp 7432–743https://doi.org/10.1609/AAAI.V34I05.6239,

Black S, Gao L, Wang P, et al (2022) Gpt-neo: Large scale autoregressive language modeling with meshtensorflow, 2021. URL: https://doiorg/105281/zenodo5297715

Borkan D, Dixon L, Sorensen J, et al (2019) Nuanced metrics for measuring unintended bias with real data for text classification. In: Companion Proceedings of The 2019 World Wide Web Conference. Association for Computing Machinery, New York, NY, USA, WWW '19, p 491–500, https://doi.org/10.1145/3308560.3317593,

Borkar J (2023) What can we learn from data leakage and unlearning for law? Preprint at arXiv:2307.10476

Bourtoule L, Chandrasekaran V, Choquette-Choo CA, et al (2021) Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, pp 141–159

Carlini N, Tramer F, Wallace E, et al (2021) Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21), pp 2633–2650

Cettolo M, Niehues J, Stüker S, et al (2014) Report on the 11th IWSLT evaluation campaign. In: Federico M, Stüker S, Yvon F (eds) Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign@IWSLT 2014, Lake Tahoe, CA, USA, December 4-5, 2014, https://aclanthology.org/2014.iwslt-evaluation.1

Chen J, Yang D (2023) Unlearn what you want to forget: Efficient unlearning for llms. Preprint at arXiv:2310.20150

Clark K, Luong M, Le QV, et al (2020) ELECTRA: pre-training text encoders as discriminators rather than generators. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, https://openreview.net/forum?id=r1xMH1BtvB

Clark P, Cowhey I, Etzioni O, et al (2018) Think you have solved question answering? try arc, the AI2 reasoning challenge. CoRR abs/1803.05457. arXiv:1803.05457

Computer T (2023) Redpajama: an open dataset for training large language models. https://github.com/togethercomputer/RedPajama-Data

Daheim N, Dziri N, Sachan M, et al (2023) Elastic weight removal for faithful and abstractive dialogue generation. Preprint at arXiv:2303.17574

De-Arteaga M, Romanov A, Wallach HM, et al (2019) Bias in bios: A case study of semantic representation bias in a high-stakes setting. In: danah boyd, Morgenstern JH (eds) Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019. ACM, pp 120–128, https://doi.org/10.1145/3287560.3287572, https://doi.org/10.1145/3287560.3287572

Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint at arXiv:1810.04805

Devlin J, Chang M, Lee K, et al (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, pp 4171–418https://doi.org/10.18653/V1/N19-1423

Dinan E, Roller S, Shuster K, et al (2019) Wizard of wikipedia: Knowledge-powered conversational agents. In: International Conference on Learning Representations

Eldan R, Russinovich M (2023) Who's harry potter? approximate unlearning in llms. Preprint at arXiv:2310.02238

Commission European (2019) Ethics guidelines for trustworthy AI. Publications Office of the EU, doi/. https://doi.org/10.2759/346720

European Parliament (2024) European parliament legislative resolution of 13 march 2024 on the proposal for a regulation of the european parliament and of the council on laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (com(2021)0206 – c9-0146/2021 – 2021/0106(cod)). https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.html

European Parliament, Council of the European Union (2016) Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://data.europa.eu/eli/reg/2016/679/oj

Fan A, Lewis M, Dauphin YN (2018) Hierarchical neural story generation. In: Gurevych I, Miyao Y (eds) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers. Association for Computational Linguistics, pp 889–89https://doi.org/10.18653/V1/P18-1082, https://aclanthology.org/P18-1082/

Gallegos IO, Rossi RA, Barrow J, et al (2024) Self-debiasing large language models: Zero-shot recognition and reduction of stereotypes. Preprint at arXiv:2402.01981

Gao L, Biderman S, Black S, et al (2021) The pile: An 800gb dataset of diverse text for language modeling. CoRR abs/2101.00027. arXiv:2101.00027

Gehman S, Gururangan S, Sap M, et al (2020) Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In: Findings

Gliwa B, Mochol I, Biesek M, et al (2019) Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. CoRR abs/1911.12237. arXiv:1911.12237

Gordon A, Kozareva Z, Roemmele M (2012) SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In: Agirre E, Bos J, Diab M, et al (eds) *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). Association for Computational Linguistics, Montréal, Canada, pp 394–398, https://aclanthology.org/S12-1052

Group of Seven (G7) (2023) Hiroshima process international guiding principles for organizations developing advanced ai systems. https://www.mofa.go.jp/files/100573471.pdf

Hassan F, Sánchez D, Soria-Comas J, et al (2019) Automatic anonymization of textual documents: detecting sensitive information via word embeddings. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, pp 358–365

Hassan F, Sá¡nchez D, Domingo-Ferrer J (2023) Utility-preserving privacy protection of textual documents via word embeddings. IEEE Transactions on Knowledge and Data Engineering 35(1):1058–1071

Houlsby N, Giurgiu A, Jastrzebski S, et al (2019) Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning, PMLR, pp 2790–2799

Ilharco G, Ribeiro MT, Wortsman M, et al (2022) Editing models with task arithmetic. Preprint at arXiv:2212.04089

Jang J, Yoon D, Yang S, et al (2022) Knowledge unlearning for mitigating privacy risks in language models. Preprint at arXiv:2210.01504

Ji J, Liu M, Dai J, et al (2023) Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. CoRR abs/2307.04657. https://doi.org/10.48550/ARXIV.2307.04657, arXiv:2307.04657

Jin Q, Dhingra B, Liu Z, et al. (2019) Pubmedqa: A dataset for biomedical research question answering. In: Inui K, Jiang J, Ng V, et al. (eds) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019. Association for Computational Linguistics, pp 2567–2577, https://doi.org/10.18653/V1/D19-1259

Joseph R. Biden Jr. (2023) Executive order on the safe, secure, and trustworthy development and use of artificial intelligence. https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/

Kadhe S, Halimi A, Rawat A, et al (2023) Fairsisa: Ensemble post-processing to improve fairness of unlearning in llms. In: Socially Responsible Language Modelling Research

Kandpal N, Wallace E, Raffel C (2022) Deduplicating training data mitigates privacy risks in language models. arXiv:2202.06539

Klimt B, Yang Y (2004) Introducing the enron corpus. In: CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA, http://www.ceas.cc/papers-2004/168.pdf

Komeili M, Shuster K, Weston J (2022) Internet-augmented dialogue generation. In: Muresan S, Nakov P, Villavicencio A (eds) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Dublin, Ireland, pp 8460–8478, https://doi.org/10.18653/v1/2022.acl-long.579, https://aclanthology.org/2022.acl-long.579

Kumar VB, Gangadharaiah R, Roth D (2022) Privacy adhering machine un-learning in nlp. Preprint at arXiv:2212.09573

Lan Z, Chen M, Goodman S, et al (2020) ALBERT: A lite BERT for self-supervised learning of language representations. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, https://openreview.net/forum?id=H1eA7AEtvS

Leong CT, Cheng Y, Wang J, et al (2023) Self-detoxifying language models via toxification reversal. Preprint at arXiv:2310.09573

Levy O, Seo M, Choi E, et al (2017) Zero-shot relation extraction via reading comprehension. In: Levy R, Specia L (eds) Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017. Association for Computational Linguistics, pp 333–342, https://doi.org/10.18653/V1/K17-1034

Li J, Cheng X, Zhao X, et al (2023a) Halueval: A large-scale hallucination evaluation benchmark for large language models. In: Bouamor H, Pino J, Bali K (eds) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023. Association for Computational Linguistics, pp 6449–6464, https://aclanthology.org/2023.emnlp-main.397

Li Y, Bubeck S, Eldan R, et al (2023b) Textbooks are all you need ii: phi-1.5 technical report. Preprint at arXiv:2309.05463

Limisiewicz T, Mareček D (2022) Don't forget about pronouns: Removing gender bias in language models without losing factual gender information. Preprint at arXiv:2206.10744

Lin S, Hilton J, Evans O (2022) Truthfulqa: Measuring how models mimic human falsehoods. In: Muresan S, Nakov P, Villavicencio A (eds) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. Association for Computational Linguistics, pp 3214–3252, https://doi.org/10.18653/V1/2022.ACL-LONG.229,
Lison P, Pilán I, Sánchez D, et al (2021) Anonymisation models for text data: State of the art, challenges and future directions. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 4188–4203
Liu P, Yuan W, Fu J et al (2023) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comp Surv 55(9):1–35
Liu Y, Ott M, Goyal N, et al (2019) Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692. arXiv:1907.11692
Liu Z, Kalinli O (2023) Forgetting private textual sequences in language models via leave-one-out ensemble. Preprint at arXiv:2309.16082
Lu X, Welleck S, Hessel J et al (2022) Quark: controllable text generation with reinforced unlearning. Adv Neural Inf Process Syst 35:27591–27609
Maas AL, Daly RE, Pham PT, et al (2011) Learning word vectors for sentiment analysis. In: Lin D, Matsumoto Y, Mihalcea R (eds) The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA. The Association for Computer Linguistics, pp 142–150, https://aclanthology.org/P11-1015/
Madaan A, Tandon N, Clark P, et al (2022) Memory-assisted prompt editing to improve gpt-3 after deployment. Preprint at arXiv:2201.06009
Meng K, Bau D, Andonian A, et al (2022) Locating and editing factual knowledge in GPT. CoRR abs/2202.05262. arXiv:2202.05262
Merity S, Xiong C, Bradbury J, et al (2016) Pointer sentinel mixture models. arXiv:1609.07843
Mitchell E, Lin C, Bosselut A, et al (2022) Memory-based model editing at scale. In: International Conference on Machine Learning, PMLR, pp 15817–15831
Nadeem M, Bethke A, Reddy S (2021) StereoSet: Measuring stereotypical bias in pretrained language models. In: Zong C, Xia F, Li W, et al (eds) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, pp 5356–5371, https://doi.org/10.18653/v1/2021.acl-long.416, https://aclanthology.org/2021.acl-long.416
Nangia N, Vania C, Bhalerao R, et al (2020) CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In: Webber B, Cohn T, He Y, et al (eds) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, pp 1953–1967, https://doi.org/10.18653/v1/2020.emnlp-main.154, https://aclanthology.org/2020.emnlp-main.154
Nasr M, Carlini N, Hayase J, et al (2023) Scalable extraction of training data from (production) language models. Preprint at arXiv:2311.17035
Nguyen TT, Huynh TT, Nguyen PL, et al (2022) A survey of machine unlearning. Preprint at arXiv:2209.02299
Ni S, Chen D, Li C, et al (2023) Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. Preprint at arXiv:2311.08011
Nivre J, de Marneffe M, Ginter F, et al (2020) Universal dependencies v2: An evergrowing multilingual treebank collection. In: Calzolari N, Béchet F, Blache P, et al (eds) Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020. European Language Resources Association, pp 4034–4043, https://aclanthology.org/2020.lrec-1.497/
Paperno D, Kruszewski G, Lazaridou A, et al (2016) The LAMBADA dataset: Word prediction requiring a broad discourse context. In: Erk K, Smith NA (eds) Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp 1525–1534, https://doi.org/10.18653/v1/P16-1144, https://aclanthology.org/P16-1144
Papernot N, Song S, Mironov I, et al (2018) Scalable private learning with pate. Preprint at arXiv:1802.08908
Patil V, Hase P, Bansal M (2023) Can sensitive information be deleted from llms? objectives for defending against extraction attacks. Preprint at arXiv:2309.17410
Pawelczyk M, Neel S, Lakkaraju H (2023) In-context unlearning: Language models as few shot unlearners. Preprint at arXiv:2310.07579
Pochinkov N, Schoots N (2023) Dissecting large language models. In: Socially Responsible Language Modelling Research
Qu Y, Yuan X, Ding M, et al (2023) Learn to unlearn: A survey on machine unlearning. Preprint at arXiv:2305.07512

Radford A, Narasimhan K, Salimans T, et al (2018) Improving language understanding by generative pre-training. https://openai.com/index/language-unsupervised/
Radford A, Wu J, Child R et al (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
Raffel C, Shazeer N, Roberts A et al (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551
Rashkin H, Smith EM, Li M, et al (2019) Towards empathetic open-domain conversation models: A new benchmark and dataset. In: Korhonen A, Traum D, Màrquez L (eds) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp 5370–5381, https://doi.org/10.18653/v1/P19-1534, https://aclanthology.org/P19-1534
Ravfogel S, Elazar Y, Gonen H, et al (2020) Null it out: Guarding protected attributes by iterative nullspace projection. Preprint at arXiv:2004.07667
Rowling JK (2000) Harry Potter and the Sorcerer's Stone. Bloomsbury
Rudinger R, Naradowsky J, Leonard B, et al (2018) Gender bias in coreference resolution. In: Walker M, Ji H, Stent A (eds) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 8–14, https://doi.org/10.18653/v1/N18-2002, https://aclanthology.org/N18-2002
Sakaguchi K, Bras RL, Bhagavatula C, et al (2020) Winogrande: An adversarial winograd schema challenge at scale. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, pp 8732–8740, https://doi.org/10.1609/AAAI.V34I05.6399
Salem A, Zhang Y, Humbert M, et al (2018) Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. Preprint at arXiv:1806.01246
Sánchez D, Batet M (2016) C-sanitized: a privacy model for document redaction and sanitization. J Assoc Inf Sci Tech 67(1):148–163
Sanh V, Debut L, Chaumond J, et al (2019) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. Preprint at arXiv:1910.01108
Scao TL, Fan A, Akiki C, et al (2022) Bloom: A 176b-parameter open-access multilingual language model. Preprint at arXiv:2211.05100
Sellam T, Das D, Parikh AP (2020) BLEURT: learning robust metrics for text generation. In: Jurafsky D, Chai J, Schluter N, et al (eds) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, pp 7881–7892, https://doi.org/10.18653/V1/2020.ACL-MAIN.704
Shaik T, Tao X, Xie H, et al (2023) Exploring the landscape of machine unlearning: A survey and taxonomy. Preprint at arXiv:2305.06360
Shi W, Ajith A, Xia M, et al (2023) Detecting pretraining data from large language models. Preprint at arXiv:2310.16789
Shokri R, Stronati M, Song C, et al (2017) Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP), IEEE, pp 3–18
Silveira N, Dozat T, de Marneffe MC, et al (2014) A gold standard dependency corpus for English. In: Calzolari N, Choukri K, Declerck T, et al (eds) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland, pp 2897–2904, http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf
Smith EM, Williamson M, Shuster K, et al (2020) Can you put it all together: Evaluating conversational agents' ability to blend skills. In: Jurafsky D, Chai J, Schluter N, et al (eds) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp 2021–2030, https://doi.org/10.18653/v1/2020.acl-main.183, https://aclanthology.org/2020.acl-main.183
Smith V, Shamsabadi AS, Ashurst C, et al (2023) Identifying and mitigating privacy risks stemming from language models: A survey. Preprint at arXiv:2310.01424
Socher R, Perelygin A, Wu J, et al (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Yarowsky D, Baldwin T, Korhonen A, et al (eds) Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp 1631–1642, https://aclanthology.org/D13-1170
Stanovsky G, Smith NA, Zettlemoyer L (2019) Evaluating gender bias in machine translation. In: Korhonen A, Traum D, Màrquez L (eds) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp 1679–1684, https://doi.org/10.18653/v1/P19-1164, https://aclanthology.org/P19-1164

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: International conference on machine learning, PMLR, pp 3319–3328
Sá¡nchez D, Batet M, (2017) Toward sensitive document release with privacy guarantees. Engineering Applications of Artificial Intelligence 59:23–34
Taylor R, Kardas M, Cucurull G, et al (2022) Galactica: A large language model for science. Preprint at arXiv:2211.09085
Touvron H, Lavril T, Izacard G, et al (2023a) Llama: Open and efficient foundation language models. arXiv:2302.13971
Touvron H, Martin L, Stone K, et al (2023b) Llama 2: Open foundation and fine-tuned chat models. efficient foundation language models. arXiv:2307.09288
Tuggener D, von Däniken P, Peetz T, et al (2020) LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In: Calzolari N, Béchet F, Blache P, et al (eds) Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020. European Language Resources Association, pp 1235–1241, https://aclanthology.org/2020.lrec-1.155/
Tunstall L, Von Werra L, Wolf T (2022) Natural language processing with transformers. " O'Reilly Media, Inc."
United Nations (1948) Universal Declaration of Human Rights
Üstün A, Aryabumi V, Yong ZX, et al (2024) Aya model: An instruction finetuned open-access multilingual language model. Preprint at arXiv:2402.07827
Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30
Wang L, Chen T, Yuan W, et al (2023) Kga: A general machine unlearning framework based on knowledge gap alignment. Preprint at arXiv:2305.06535
Wang L, Zeng X, Guo J, et al (2024) Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. Preprint at arXiv:2402.05813
Webster K, Recasens M, Axelrod V et al (2018) Mind the GAP: a balanced corpus of gendered ambiguous pronouns. Trans Assoc Comput Linguistics 6:605–617 https://doi.org/10.1162/tacl_a_00240, URL https://aclanthology.org/Q18-1042
Wu X, Li J, Xu M, et al (2023) Depn: Detecting and editing privacy neurons in pretrained language models. Preprint at arXiv:2310.20138
Xu H, Zhu T, Zhang L et al (2023) Machine unlearning: a survey. ACM Comp Surv 56(1):1–36
Yao Y, Xu X, Liu Y (2023) Large language model unlearning. Preprint at arXiv:2310.10683
Yeom S, Giacomelli I, Fredrikson M, et al (2018) Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF), IEEE, pp 268–282
Yoon D, Jang J, Kim S, et al (2023) Gradient ascent post-training enhances language model generalization. Preprint at arXiv:2306.07052
Yu C, Jeoung S, Kasi A et al (2023) Unlearning bias in language models by partitioning gradients. Find Assoc Comput Linguistics: ACL 2023:6032–6048
Zellers R, Holtzman A, Bisk Y, et al (2019) Hellaswag: Can a machine really finish your sentence? CoRR abs/1905.07830. arXiv:1905.07830
Zhang J, Liu J, He J, et al (2024a) Composing parameter-efficient modules with arithmetic operation. Advances in Neural Information Processing Systems 36
Zhang N, Yao Y, Tian B, et al (2024b) A comprehensive study of knowledge editing for large language models. Preprint at arXiv:2401.01286
Zhang S, Dinan E, Urbanek J, et al (2018) Personalizing dialogue agents: I have a dog, do you have pets too? In: Gurevych I, Miyao Y (eds) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, pp 2204–2213https://doi.org/10.18653/v1/P18-1205, https://aclanthology.org/P18-1205
Zhang S, Roller S, Goyal N, et al (2022) Opt: Open pre-trained transformer language models. arXiv:2205.01068
Zhang X, Zhao JJ, LeCun Y (2015) Character-level convolutional networks for text classification. In: Cortes C, Lawrence ND, Lee DD, et al (eds) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp 649–657, https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html
Zhao J, Wang T, Yatskar M, et al (2018) Gender bias in coreference resolution: Evaluation and debiasing methods. In: Walker M, Ji H, Stent A (eds) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 15–20, https://doi.org/10.18653/v1/N18-2003, https://aclanthology.org/N18-2003

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Alberto Blanco-Justicia[1] · Najeeb Jebreel[1] · Benet Manzanares-Salor[1] · David Sánchez[1] · Josep Domingo-Ferrer[1] · Guillem Collell[2] · Kuan Eeik Tan[2]

✉ David Sánchez
david.sanchez@urv.cat

Alberto Blanco-Justicia
alberto.blanco@urv.cat

Najeeb Jebreel
najeeb.jebreel@urv.cat

Benet Manzanares-Salor
benet.manzanares@urv.cat

Josep Domingo-Ferrer
josep.domingo@urv.cat

Guillem Collell
guillem.collell@huawei.com

Kuan Eeik Tan
kuan.eeik.tan@huawei.com

1 Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Av. Països Catalans 26, Catalonia, E-43007 Tarragona, Spain

2 Huawei Technologies Finland Research Center, Itämerenkatu 9, FI-00180 Helsinki, Finland