Rethinking Evaluation Methods for Machine Unlearning

Leon Wichert
Leibniz University Hannover
leon.wichert@gmail.com

Sandipan Sikdar
L3S Research Center
Leibniz University Hannover
sandipan.sikdar@l3s.de

Abstract

Machine *unlearning* refers to methods for deleting information about specific training instances from a trained machine learning model. This enables models to delete user information and comply with privacy regulations. While retraining the model from scratch on the training set excluding the instances to be "*forgotten*" would result in a desired unlearned model, owing to the size of datasets and models, it is infeasible. Hence, unlearning algorithms have been developed, where the goal is to obtain an unlearned model that behaves as closely as possible to the retrained model. Consequently, evaluating an unlearning method involves - (i) randomly selecting a *forget* set (i.e., the training instances to be unlearned), (ii) obtaining an unlearned and a retrained model, and (iii) comparing the performance of the unlearned and the retrained model on the test and forget set. However, when the forget set is randomly selected, the unlearned model is almost often similar to the original (i.e., prior to unlearning) model. Hence, it is unclear if the model did really unlearn or simply copied the weights from the original model. For a more robust evaluation, we instead propose to consider training instances with significant influence on the trained model. When such influential instances are considered in the forget set, we observe that the unlearned model deviates significantly from the retrained model. Such deviations are also observed when the size of the forget set is increased. Lastly, choice of dataset for evaluation could also lead to misleading interpretation of results.

1 Introduction

Datasets that are used to train natural language processing models are often very large and may contain sensitive information, which raises concerns regarding user privacy (Shaik et al., 2023). To address these concerns, the "right to be forgotten" has been introduced and realized through laws such as the General Data Protection Regulation and the California Consumer Privacy Act. This right provides users with the ability to request the removal of their personal data from a machine learning pipeline. While personal information can simply be deleted from a database, this is much more complicated for the training data in machine learning, as the final trained model still encapsulates the user's information, posing a potential risk for data leakage (Nguyen et al., 2022). To fully comply with the law, not only the data point but also its influence on the trained model needs to be removed. This is realized through machine unlearning techniques, which remove the influence of specific points in efficient ways without compromising model performance. Note that a trivial solution to the problem would be to retrain the model from scratch. However, this would be infeasible for larger models, which involve long training times. Nevertheless, the goal of the unlearning approaches is to achieve an unlearned model that behaves as closely as possible to a retrained model. While machine unlearning research has gained traction in recent years, the focus has been mainly on computer vision tasks (Bourtoule et al., 2021; Guo et al., 2020; Mehta et al., 2022). For models involving textual data, only a few unlearning approaches (Kumar et al., 2023; Wang et al., 2023) exist.

Typically, an unlearning method is evaluated on its ability to replicate the performance of the retrained model on the test set and the forget set (i.e., the set of points whose influence is to be removed). However, the forget set is usually *small* and *randomly sampled* from the training set. We argue that a small, randomly selected forget set lacks the necessary complexity to effectively evaluate the unlearning process and is not representative of the true performance of the unlearning method. This is illustrated in Table 1, where we perform a weight comparison for a state-of-the-art unlearning algorithm (Wang et al., 2023). The

Findings of the Association for Computational Linguistics: EACL 2024, pages 4727–4739

original model is a DistilBERT (Sanh et al., 2019) model fine-tuned to perform text classification on the LEDGAR (Tuggener et al., 2020) dataset. For unlearning evaluation, 100 instances are randomly selected, and the unlearning algorithm is applied to obtain the unlearned model. We compare the weights of the unlearned model to those of the original model and observe that they are almost identical. It is hence unclear if the forget set didn't have any influence on the model or if the unlearning was unsuccessful.

Table 1: Weight comparisons **with the original model** for KGA unlearning on the LEDGAR dataset. The model architecture is DistilBERT, and 100 instances are randomly selected as the forget set. We flattened the parameters for each layer group and calculated the cosine similarity. The unlearned model's weights are nearly identical to those of the original model.

Layers	Retrained Model	Unlearned Model
Embedding	0.999904	1.000000
Transformer Block 1	0.998690	1.000000
Transformer Block 2	0.998532	1.000000
Transformer Block 3	0.998730	1.000000
Transformer Block 4	0.998912	1.000000
Transformer Block 5	0.998834	1.000000
Transformer Block 6	0.998725	1.000000
Classifier	0.001149	0.999999

For a more robust evaluation, we propose an alternative evaluation method where the forget sets consist of highly influential data points, which we identify using influence functions (Koh and Liang, 2017). These influential data points act as representatives of sensitive user information, allowing us to better gauge the algorithm's success in unlearning tailored to practical scenarios. Additionally, we also consider larger forget sets and different datasets for a more robust evaluation.

Our results show that influential forget sets created via influence functions provide a more challenging unlearning scenario for state-of-the-art machine unlearning methods, as they are unable to match the behavior of the retrained model. Moreover, we observe that the impact of influential data varies across different datasets, showing the importance of dataset choice for machine unlearning evaluation. Finally, we demonstrate that increasing the size of the forget set improves the robustness of the evaluation for both random and influential forget data, which is demonstrated by the drop in performance of the unlearned models for larger forget sets even though the model retrained from scratch maintains its performance.

To summarize the primary contributions of our work -

- we point out the inability of existing evaluation methods for unlearning in demonstrating their true performance
- we propose an alternate evaluation approach involving influential function for a more representative evaluation of unlearning methods
- our findings provide recommendations for a more robust evaluation
- to facilitate reproducibility, we make project-related resources publicly available[1].

2 Background and related work

In this section, we formally define the machine unlearning problem and discuss existing machine unlearning methods, focusing on their applications in natural language processing. Additionally, we describe the Knowledge Gap Alignment (KGA) (Wang et al., 2023) in detail, which is the state-of-the-art unlearning algorithm for natural language processing and which we deploy for our experiments.

2.1 The unlearning problem

Given a model A_D trained on the training data D and the forget dataset $D_f \subset D$, an unlearning algorithm is denoted as a function $U(A_D, D, D_f)$ that outputs a new model A^*, which crucially maintains the performance on $D_r = D \setminus D_f$ (Nguyen et al., 2022). The training instances containing the information that should be removed are commonly referred to as the forget set, and unlearning aims to remove their influence from the already trained model. A simple approach to the machine unlearning problem can be realized by deleting the forget data and then retraining the model on the remaining data D_r. This ensures a complete removal of the data in question while maintaining the performance in the best way possible, as the model is optimized during retraining. However, this approach is considered impractical for large-scale models due to the significant time costs associated with it (Bourtoule et al., 2021).

[1] https://github.com/Kartoffelpuffa/Rethinking-MU-Evaluation

2.2 Exact and approximate unlearning

According to Thudi et al. (2022), machine unlearning algorithms can further be categorized into two categories: exact and approximate unlearning. Exact unlearning methods ensure the complete removal of the forget set, as they rely on retraining from scratch, but they improve the efficiency of this retraining process. A large portion of exact unlearning methods are based on SISA (Bourtoule et al., 2021), which divides the training data into multiple disjoint shards. On each shard, a model is trained in isolation, and for prediction, different aggregation strategies are used. Upon receiving an unlearning request, only the models trained on instances from the forget set need to be retrained, reducing the unlearning time. However, since SISA requires a lot of storage for the additional models, it is impractical for modern natural language processing. Kumar et al. (2023) combat this by introducing strategies to reduce the number of parameters that need to be stored, enabling the use of the SISA algorithm for natural language processing tasks. It's important to note that since aggregation is required, the approach cannot be used for generative tasks. This limitation is addressed by approximate unlearning techniques, which do not involve any form of retraining. Instead, these methods modify the parameters of the original model directly to obtain an unlearned model that performs as close as possible to a retrained model. Often, such algorithms involve the calculation of influence of instances in the forget set and reversing this influence during unlearning (Guo et al., 2020; Mehta et al., 2022). As the calculation of these influence scores can be costly, especially for larger models, an alternative solution is proposed by Chundawat et al. (2023), who perform unlearning based on the assumption that the unlearned model should obtain random performance regarding the forget set, which is achieved through knowledge adaptation from a random teacher model.

2.3 Knowledge Gap Alignment (KGA)

In the domain of natural language processing, exploring and developing unlearning techniques have largely remained unexplored. However, the Knowledge Gap Alignment (KGA) (Wang et al., 2023) framework has shown superior results compared to previous approximate unlearning techniques. The key assumption behind KGA is that an unlearned model should treat the forget data the same as previously unseen data, which is how a retrained model would handle it. KGA achieves this via *knowledge gaps*, defined as the distance between prediction distributions from two models with the same architecture but different training data.

The algorithm operates on three datasets: the original training data D, the forget data D_f, and an extra dataset D_n that is distinct from D but has a similar distribution. During the unlearning, the original model A_D, trained on D, is transformed into the unlearned model A^*. Directly aligning the prediction distribution of A^* on D_f with the prediction distribution of A_D on D_n might be challenging because D_n could contain unknown labels or features. Instead, the *knowledge gap* between A_D and A_n on D_n is aligned with the *knowledge gap* between A^* and A_f on D_f using Kullback-Leibler divergence as follows:

$$\mathcal{L}_a = \sum_{(y,z)\in(D_f,D_n)} |KL[Pr_{(A^*)}(y)||Pr_{(A_f)}(y)] - KL[Pr_{(A_D)}(z)||Pr_{(A_n)}(z)]|, \quad (1)$$

where $Pr_{(A)}(z)$ refers to the output distribution of model A given input z. The loss is summed over batches containing pairs of instances (y, z), which are sampled from (D_f, D_n). Meanwhile, the *knowledge gap* between A^* and A_D on D_r is minimized in order to maintain the performance on the remaining data:

$$\mathcal{L}_r = \sum_{x\in D_r} KL[Pr_{(A^*)}(x)||Pr_{(A_D)}(x)]. \quad (2)$$

Both objectives are then optimized together as follows:

$$\mathcal{L} = \mathcal{L}_a + \alpha \cdot \mathcal{L}_r. \quad (3)$$

Due to its state-of-the-art performance on different natural language processing tasks, we consider KGA a representative approximate unlearning technique for our experiments.

3 Method

In this section, we discuss the challenges of the current evaluation strategy and proceed to introduce our approach involving influential forget datasets based on influence functions (Koh and Liang, 2017).

3.1 The Need for Robust Evaluation

For evaluation of machine unlearning techniques, the unlearned model is compared to a retrained model using a randomly created forget set. Following this procedure, we perform evaluation of the KGA method on the LEDGAR (Tuggener et al., 2020) text classification task using a fine-tuned DistilBERT (Sanh et al., 2019) as the original model. While the resulting F1 scores on the test and forget set are high (Table 2), we also notice that the obtained unlearned model is very similar to the original model with respect to its weights, as seen in Table 1. This is further supported by a comparison of the prediction distributions using Jensen-Shannon divergence, as depicted in Table 3.

Table 2: Performance comparisons for KGA unlearning. The model architecture is DistilBERT, and 100 instances are randomly selected as the forget set. The unlearned model obtains high performance on test and forget sets.

Model	**Test**	**Forget**
	F1 Score (%)	F1 Score (%)
Original	94.84	93.81
Retrained	94.79	93.27
Unlearned	94.88	94.69

Table 3: Output distribution comparisons for KGA unlearning using Jensen-Shannon divergence (JSD). The model architecture is DistilBERT, and 100 instances are randomly selected as the forget set. The output distributions of the original and unlearned models are very similar.

Comparison	**Test**	**Forget**
	JSD ↓	JSD ↓
Retrained vs Unlearned	0.0094	0.0235
Original vs Retrained	0.0094	0.0058
Original vs Unlearned	0.0005	0.0085

Therefore, even though the unlearned model is able to match the performance of the retrained model, the evaluation does not provide significant insights, as the unlearning technique resulted in a model with minimal changes. In order to provide a more robust evaluation, we experiment with forget sets of different sizes and varying degrees of influence.

3.2 Selecting Influential Forget Data

While changing the size of the forget set is straightforward, the meaning of influence needs to be further specified. We consider an influential forget set to contain instances which have a significant influence on the original model and calculate these instances using influence functions (Koh and Liang, 2017). The core idea behind them is to think of the influence of a point z on a given model as how the model's parameters would change if z were removed from the training data. As it is not feasible to observe these parameter changes by retraining the model for every point z, the authors instead upweight z by a small amount in order to create similar parameter changes. These changes can be calculated and are defined as the influence of z on the model parameters:

$$\mathcal{I}_{up,params}(z) = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}) \quad (4)$$

Furthermore, the influence of z on a specific test point z_{test} can be derived by application of the chain rule, as the loss of z_{test} is a function of the parameters:

$$\mathcal{I}_{up,loss}(z, z_{test}) = - \nabla_{\theta} L(z_{test}, \hat{\theta})^{\top} H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}) \quad (5)$$

The calculations required for influence functions involve the computation of the inverse Hessian of the loss function $H_{\hat{\theta}}^{-1}$, which is computationally expensive and scales poorly with the number of parameters. Instead of calculating and inverting the Hessian, the authors also propose to calculate a Hessian-vector product directly, which does not require the full Hessian and is achievable in linear time with respect to the number of parameters. The inverse Hessian-vector product can then be estimated through the Linear time Stochastic Second-Order Algorithm (LiSSA) (Agarwal et al., 2017) in a recursive manner for T steps.

An alternative method for calculating influence was proposed by Pruthi et al. (2020), which involves monitoring changes in test loss throughout the training process. However, scaling this method for large natural language processing models requires the storage of multiple checkpoints, resulting in substantial memory demands that were impractical for our study.

3.3 Unlearning with Influential Forget Data

Before using the influential instances for unlearning evaluation, we need to verify the meaning of the obtained influence scores, as the application of influence functions to deep models yields different results compared to the original motivation Bae

et al. (2022). In order to determine the usefulness of the influence scores, we look at their distribution, where we expect to see varying levels of influence, so that we can select the most influential instances for unlearning evaluation. Additionally, we analyze the training performance of the original model on these influential points in comparison with random points to further verify the selected points.

To form the different forget sets for unlearning evaluation, we choose the top 1%, 2%, 5%, 10%, 15%, and 20% of the most influential points. Usually, in the existing literature, a fixed number of only 100 instances is considered, which is often less than 1% of the training data and might not be sufficient, as indicated by the lack of weight changes. While the higher percentages are less likely to occur in real-world scenarios, they still help us understand the implications of larger forget sets in machine unlearning evaluation.

We proceed to perform unlearning for each forget set, utilizing the KGA algorithm, and compare the results to retraining from scratch. Our experiments are focused on text classification, where we fine-tune DistilBERT on three different text classification datasets. Moreover, we modify the model slightly by freezing the transformer layers and focusing solely on training and unlearning the fully connected classifier layers (from now on referred to as Frozen DistilBERT). Additionally, we compare the influential method to a fully random forget set for each percentage size and carry out each individual unlearning experiment three times to account for stochasticity.

4 Experiments

In this section, we delve into the experimental setup and results of our analysis on machine unlearning evaluation.

4.1 Datasets

We conduct our experiment on three text classification datasets: IMDB (Maas et al., 2011), SST2 (Socher et al., 2013) and TREC (Li and Roth, 2002). The IMDB dataset consists of 50,000 movie reviews associated with either a negative or a positive sentiment and is the most used dataset in text-based machine unlearning-related research, according to Shaik et al. (2023). SST2 is another binary classification task based on movie reviews, including 9613 sentences. Being part of the GLUE (Wang et al., 2018) benchmark, it is a widely used dataset for evaluating text classification frameworks. The TREC dataset deals with question classification, with six different labels indicating the type of question. As the average input length is only ten tokens, it is considered an easy task for transformer-based models by Karl and Scherp (2022) and is therefore expected to provide the best results in an unlearning scenario. For each dataset, we set aside 100 instances from the training data for the extra dataset A_n required for the KGA algorithm.

We selected the text classification task for our experiments, as it was also used by the state-of-the-art KGA algorithm for evaluation. Nonetheless, our approach is easily extendable to various other natural language processing tasks, as it only necessitates a differentiable loss function, which is available for any such task.

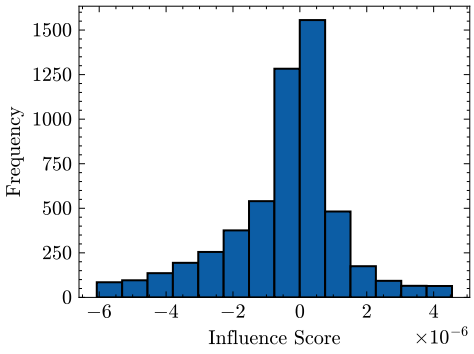
4.2 Hyperparameter Choices

To train the original models, we employ the same hyperparameters identical to the work of Wang et al. (2023) across all datasets. We use a batch size of 16 during fine-tuning. For the Frozen DistilBERT we use a learning rate of 3e-4 selected from the set of [1e-3, 5e-4, 3e-4, 1e-4, 5e-5, 3e-5]. Due to the lack of validation data for some datasets, we utilize the training loss for validation and stop the fine-tuning after 3, 3, and 6 epochs for IMDB, SST2, and TREC, respectively. Additional hyperparameters for the KGA unlearning algorithm are set according to the original paper.

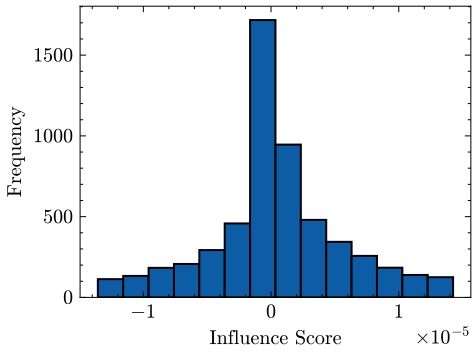
Table 4: Comparing the performance of the original model between randomly sampled and influential forget data for the SST2 dataset. Forget accuracies on influential data are significantly lower compared to random sampling, but the gap becomes smaller for larger D_f.

$\mathbf{D_f}$ (%)	DistilBERT Forget Acc. (%)		Frozen DistilBERT Forget Acc. (%)	
	Random Sampling	Influence Functions	Random Sampling	Influence Functions
1	92.30 ± 2.19	10.10	75.83 ± 4.43	0.00
2	93.97 ± 1.07	22.50	83.10 ± 3.57	0.00
5	94.60 ± 0.35	52.30	83.90 ± 2.03	1.20
10	94.83 ± 0.70	74.60	83.83 ± 0.85	26.70
15	95.60 ± 0.89	82.80	85.10 ± 1.64	50.90
20	95.40 ± 0.20	87.00	85.67 ± 1.29	63.20

For the approximation of influence functions with LiSSA (Agarwal et al., 2017), we choose the number of repetitions as $R = 1$ for the fastest computation. The recursion depth T is set to equal the size of the training data $|D|$, as the recommendation of the original influence functions paper is to have $R * T \approx |D|$. Additionally, a damping parameter $\lambda = 0.003$ and a scaling factor $\sigma = 10000$ are



(a) DistilBERT



(b) Frozen DistilBERT

Figure 1: Influence score distribution for the SST2 dataset. Scores near zero are predominant across both architectures and the shapes indicate normal distributions, with a slight shift towards negative scores for the DistilBERT model.

used for the calculations.

4.3 Influential Points

In order to verify the meaning of the influential points, we take a look at the distribution of influence scores. We illustrate our results using the SST2 dataset as a representative example. The results corresponding to the other datasets are provided in the appendix. In Figure 1, we can see that the main bulk of influential points is concentrated close to zero for the DistilBERT and Frozen DistilBERT models. Although the absolute values themselves are not particularly meaningful, the shapes indicate normal distributions, which verify the validity of these scores.

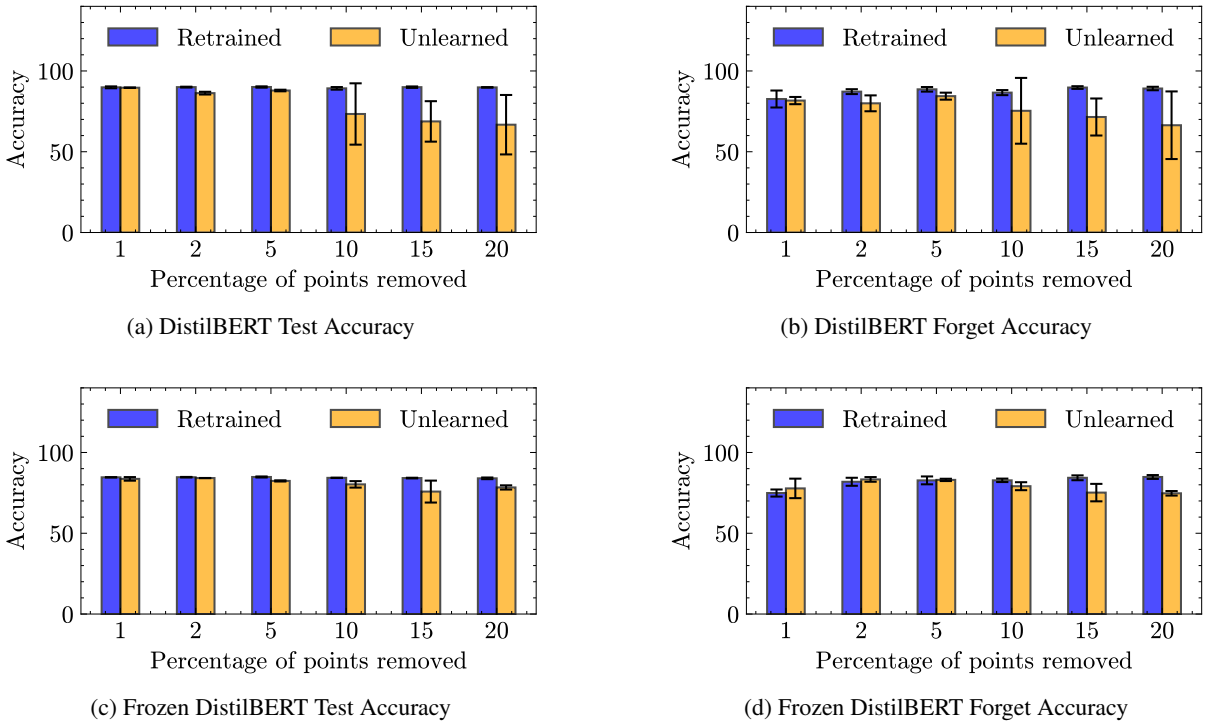
We also compare the performance of the original model between randomly sampled and influential forget data in Table 4, again using the SST2 dataset as a representative example. The accuracy on influential forget data is always lower than that on randomly sampled forget data for all sizes of D_f, showing that the influential points were harder to classify during training. When the size of the forget set increases, the accuracy on the influential data rises significantly. This complements the observations from the score distribution, as the larger sets include so many points that some of them have to be of lower influence. Overall, our influential points have a plausible distribution and a measurable effect on the original model.

4.4 Increasing Forget Set Size

For unlearning evaluation, we compare the unlearned model to a model retrained from scratch regarding test and forget accuracies. We first investigate the effects of increasing the size of the forget data while still selecting the points randomly. Figure 2 illustrates representative results using the SST2 dataset. We observe that the unlearned model performs similarly to the retrained one for sizes 1-5% but achieves worse performance for sizes 10-20% for both test and forget accuracies. The drop in performance is more noticeable for the DistilBERT model, and it is also accompanied by high standard deviations. Notably, the results for the IMDB dataset differ, as the corresponding DistilBERT model matches the performance of the retrained model across all sizes. In summary, an effect on the unlearning performance is only noticeable for higher percentages, which might not be applicable to real-world scenarios as deletion requests would not be received in such a large bulk. Nevertheless, such experiments allow for a more robust evaluation of the unlearning methods.

4.5 Unlearning Influential Data

We repeat the previous experiment but use influential points in order to create the forget sets. The results for the SST2 dataset are shown in Figure 3. Additional results on other datasets are provided in the appendix. Compared to the randomly selected forget sets, the forget accuracies behave vastly differently. In the Frozen DistilBERT architecture, they are significantly higher for the unlearned model than for the retrained model, while this difference varies for the DistilBERT model. Additionally, the forget accuracies are a lot lower compared to the test accuracies, which was not the case for the randomly selected data. These observations can also be made for IMDB and TREC, but there are some exceptions, as the forget accuracies for DistilBERT on IMDB are constantly higher



(a) DistilBERT Test Accuracy

(b) DistilBERT Forget Accuracy

(c) Frozen DistilBERT Test Accuracy

(d) Frozen DistilBERT Forget Accuracy

Figure 2: Comparison of retraining and unlearning with randomly sampled forget data D_f for the SST2 dataset. DistilBERT exhibits a significant performance drop for unlearning when $10 - 20\%$ of points are removed.

than the retrained ones, while the forget accuracies for Frozen DistilBERT on TREC align with the retrained ones.

For the test accuracies, a performance drop is noticeable for the unlearned model for sizes 10-20%. However, this only applies to the DistilBERT model, while for the frozen variant, the accuracies of the unlearned model exceed those of the retrained model at larger sizes. There is no observed performance drop for IMDB, while the unlearned model's accuracies being higher than the retrained ones is not observed for TREC.

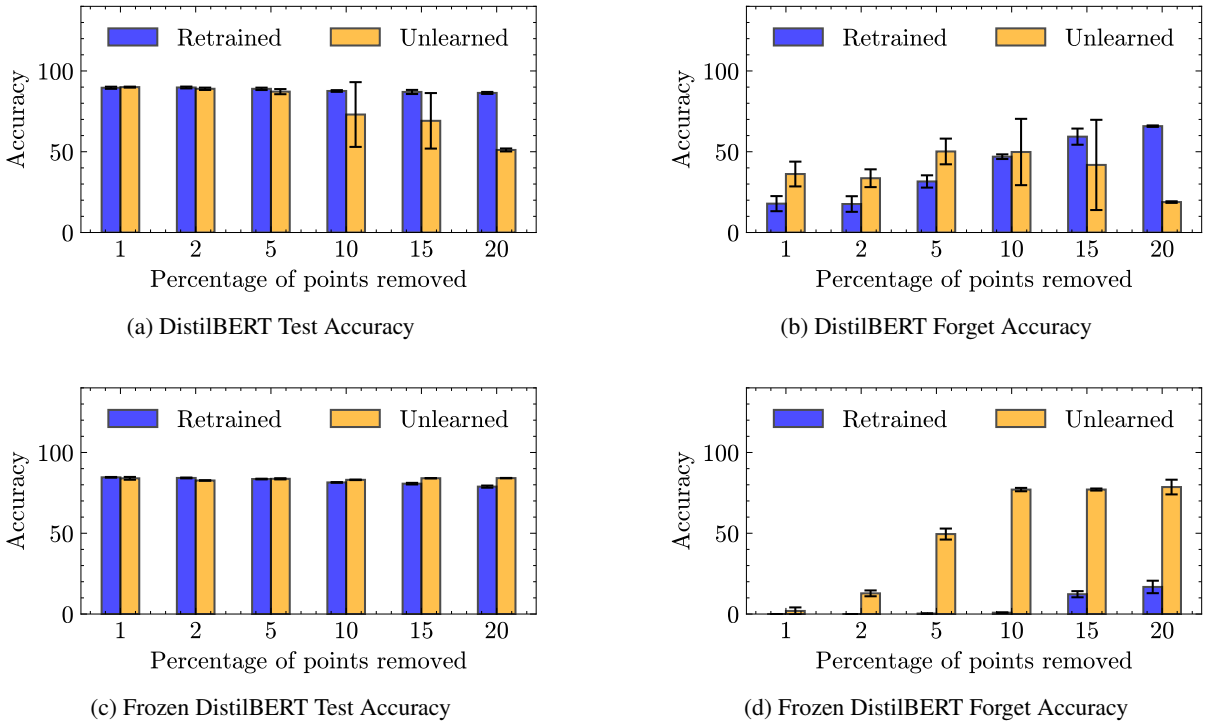
5 Discussion

In this section, we discuss the most important results from our experiments as well as their implications and our recommendations.

Calculation of Influence Scores In order to provide a more robust evaluation, we experiment with larger forget datasets, which also include influential instances determined via influence functions. The calculation of these influential points is computationally expensive for larger models and thus necessitates the use of approximations. We observe that the resulting influence scores show plausible distributions and result in forget datasets that significantly differ from random ones when evaluated by the original model. It would be intriguing to explore whether other approximation methods, such as TracIn (Pruthi et al., 2020), yield similar results and if the identified influential instances correspond to meaningful data in real-world applications. However, we were unable to deploy TracIn due to its high memory requirements.

Larger Forget Sets Using the influential points for unlearning evaluation leads to unlearned models that deviate significantly from the retrained ones, especially on the forget accuracies. On the test accuracies, the unlearned model diverges from the retrained model for sizes 10-20%. However, the same observation can be made for randomly selected forget data, showing that an increase in forget data consistently leads to worse unlearning performance. As the retrained model is able to maintain its performance even for larger forget data, this points to an area of improvement for unlearning algorithms.

Forget Accuracies on Influential Data If we consider the forget accuracies, the impact of using influential points for unlearning is more significant. Even for small forget datasets that are more relevant for real-world problems, we can see the difference between unlearned and retrained models. Moreover, even the retrained models exhibit lower forget accuracies compared with the test accuracies. This is expected, as the influential points are supposed



(a) DistilBERT Test Accuracy

(b) DistilBERT Forget Accuracy

(c) Frozen DistilBERT Test Accuracy

(d) Frozen DistilBERT Forget Accuracy

Figure 3: Comparison of retraining and unlearning with influential forget data D_f for the SST2 dataset. Test accuracies decrease for removal numbers of $10 - 20\%$ in the DistilBERT architecture. Observations for the forget accuracies diverge when comparing model architectures.

to be harder to unlearn. It is important to consider such a scenario for unlearning evaluation, as an unlearning task will not always feature points that can be easily forgotten. That the forget accuracies of the unlearned model do not align with the retrained model indicate that current algorithms are not able to effectively tackle this challenge. Moreover, this issue can not be seen when using randomly selected points, which underlines the effectiveness of the influential approach.

Choice of Dataset We have conducted our experiments on three text classification datasets. While the results for randomly selected data generalize well onto these datasets, we observe some differences when working with influential instances. This shows that we need to carefully consider the choice of dataset for machine unlearning evaluation, as different datasets can yield varying results in our influential evaluation method. Moreover, when choosing the LEDGAR dataset used in the KGA paper, we observe that removing up to 90% of the training data does not affect the performance of the retrained model significantly. This indicates that most of the data points have no significant influence on the model and removing them would not lead to any change in the model parameters. Consequently, such datasets might not be suitable for unlearning evaluation if they are not sensitive enough to the removal of instances. Further exploration of datasets is warranted to identify ones that are most relevant in an unlearning evaluation scenario.

Recommendations: Our findings lead to the following recommendations regarding the evaluation of unlearning methods:

Random and influential points. Random points provide insights into the general capabilities of an unlearning algorithm, while influential instances present more challenging scenarios that better reflect real-world applications. A goal of machine unlearning research is the creation of an evaluation benchmark (Nguyen et al., 2022) and we argue that both types of data should be included in unlearning to identify the strengths and weaknesses of different methods, which would then lead to the development of more robust and effective unlearning techniques.

Selecting datasets. We suggest using the distribution of influence scores to find useful datasets for unlearning evaluation and to create the forget sets appropriately. If a dataset contains only a few influential points or if all points have similar influence levels, the dataset is not suitable for evaluation based on influence.

Finding influential points. We note that incorporating influential points into the evaluation process introduces new challenges that must be addressed by future research. Calculating influence needs to be scalable to larger models and applicable to natural language processing tasks. Additionally, there is a need for the development of novel unlearning techniques capable of effectively handling such influential points.

6 Conclusion

In this paper, we introduce novel techniques in order to improve the robustness of machine unlearning evaluation for natural language processing. Our approach of using influential forget data created via influence functions provides a challenging unlearning scenario for state-of-the-art machine unlearning concerning the forget accuracies. Moreover, we show that increasing the size of the forget dataset also enables a more robust evaluation. Our results further demonstrate that for evaluating these unlearning methods, one needs to carefully consider the evaluation dataset. Otherwise, it might lead to incorrect interpretation of results. Our results advocate for development of unlearning evaluation scenarios resembling real-world challenges.

Limitations

Additional models. In this study, we mainly considered two variants of the DistilBERT model for our analysis. For a more robust evaluation, one could consider additional models. However, computing influence function scores for training data points is computationally very expensive, which limits its deployment to larger datasets or wider range of models. Nevertheless, we believe the observations and the key findings to hold for other datasets and models.

Additional tasks. For our analysis, we focused on the task of text classification. Our proposed method could also be extended to other tasks, including natural language generation. However, this would require adapting influence functions to the generative modeling task.

Identifying influential points. We deployed influence function to identify influential training points. However, as alluded to previously, it is computationally expensive and does not scale to large datasets. We also deployed TracIn (Pruthi et al., 2020) as an alternative method for identifying influential points. However, we also found this to be extremely resource-intensive and unable to scale to large models. Other methods for finding influential data points could be considered in the future.

Acknowledgements

This research was supported by the Federal Ministry of Education and Research by the Lower Saxony Ministry of Science and Culture (MWK) through the zukunft.niedersachsen program of the Volkswagen Foundation (HybrInt).

References

Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, 18:116:1–116:40.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B. Grosse. 2022. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 141–159. IEEE.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan S. Kankanhalli. 2023. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 7210–7217. AAAI Press.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. 2020. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR.

Fabian Karl and Ansgar Scherp. 2022. Transformers are short text classifiers: A study of inductive short text classifiers on benchmarks and real-world datasets. *CoRR*, abs/2211.16878.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings*

of Machine Learning Research, pages 1885–1894. PMLR.

Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. 2023. Privacy adhering machine un-learning in NLP. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 - Findings, Nusa Dua, Bali, November 1-4, 2023*, pages 268–277. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N. Ravi. 2022. Deep unlearning via randomized conditionally independent hessians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10412–10421. IEEE.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *CoRR*, abs/2209.02299.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Thanveer Basha Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. 2023. Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. *CoRR*, abs/2305.06360.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 4007–4022. USENIX Association.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1235–1241. European Language Resources Association.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. 2023. KGA: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13264–13276. Association for Computational Linguistics.

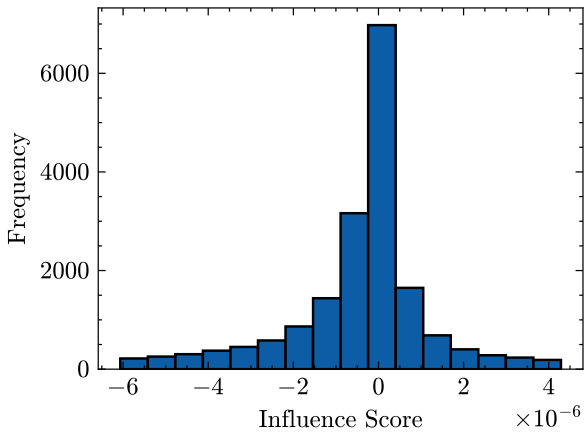
A Appendix

A.1 Additional results

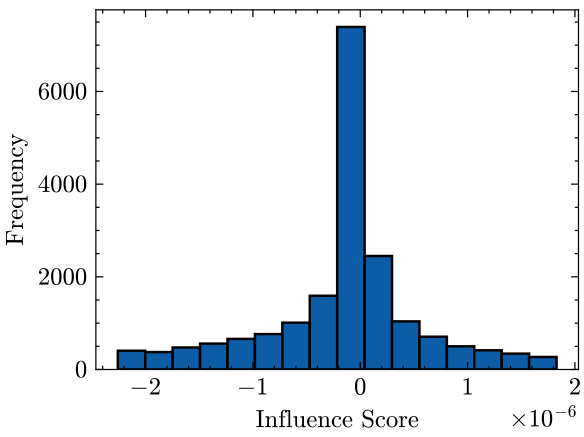
Additional results on IMDB and TREC datasets are provided here.

Table 5: Comparing the performance of the original model between randomly sampled and influential forget data for the IMDB dataset. Forget accuracies on influential data are significantly lower compared to random sampling, but the gap becomes smaller for larger D_f.

$\mathbf{D_f}$(%)	DistilBERT Forget Acc. (%)		Frozen DistilBERT Forget Acc. (%)	
	Random Sampling	Influence Functions	Random Sampling	Influence Functions
1	95.73 ± 0.83	32.80	86.13 ± 1.29	0.00
2	96.13 ± 0.46	42.60	88.20 ± 1.40	0.00
5	96.10 ± 0.44	63.80	86.13 ± 0.87	0.20
10	96.27 ± 0.31	80.70	87.83 ± 0.76	30.40
15	96.10 ± 0.35	87.00	87.00 ± 0.26	53.10
20	96.27 ± 0.12	90.20	86.70 ± 0.72	64.80

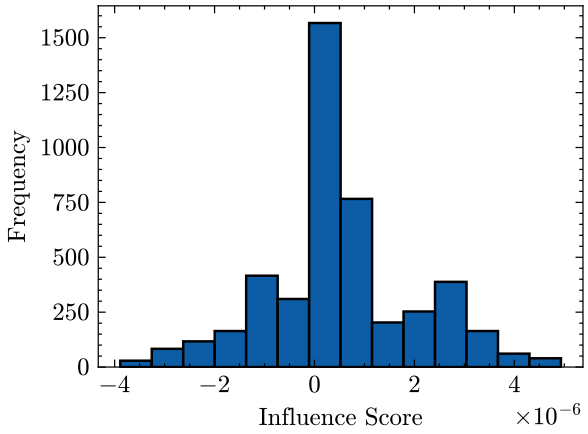


(a) DistilBERT

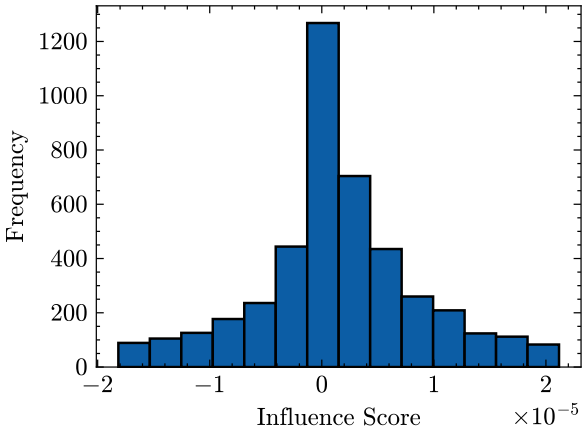


(b) Frozen DistilBERT

Figure 4: Influence score distribution for the IMDB dataset. Most of the scores are located around zero for both architectures. The scores appear to be distributed normally.



(a) DistilBERT

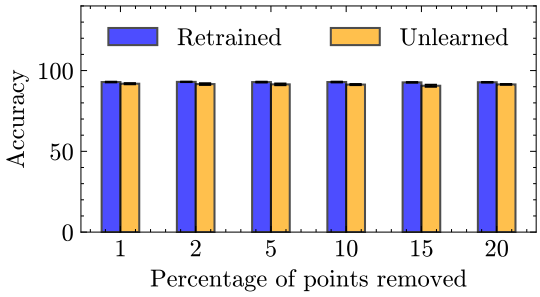


(b) Frozen DistilBERT

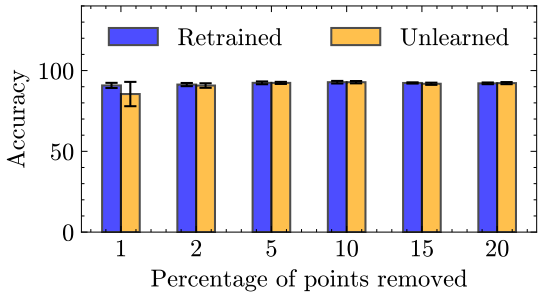
Figure 5: Influence score distribution for the TREC dataset. Most of the scores lie around zero for both architectures. The scores seem to be distributed normally, while the pattern is a little disturbed for DistilBERT.

Table 6: Comparing the performance of the original model between randomly sampled and influential forget data for the TREC dataset. Forget accuracies on influential data are significantly lower compared to random sampling, but the gap becomes smaller for larger D_f.

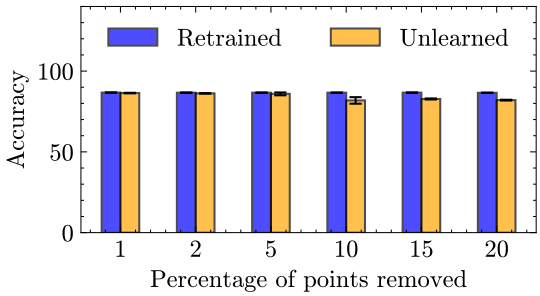
$\mathbf{D_f}$(%)	**DistilBERT Forget Acc. (%)**		**Frozen DistilBERT Forget Acc. (%)**	
	Random Sampling	Influence Functions	Random Sampling	Influence Functions
1	96.87 ± 2.14	31.50	76.53 ± 1.10	1.90
2	96.93 ± 1.10	38.50	84.73 ± 2.79	7.97
5	96.70 ± 1.10	65.80	78.70 ± 3.52	28.70
10	96.33 ± 0.76	80.00	80.50 ± 1.08	45.90
15	96.57 ± 0.12	86.70	81.00 ± 1.23	55.07
20	96.77 ± 0.50	89.90	81.33 ± 1.12	60.70



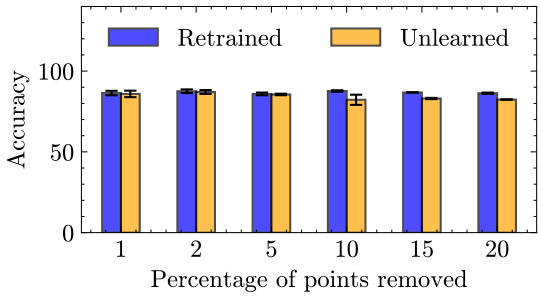
(a) DistilBERT Test Accuracy



(b) DistilBERT Forget Accuracy

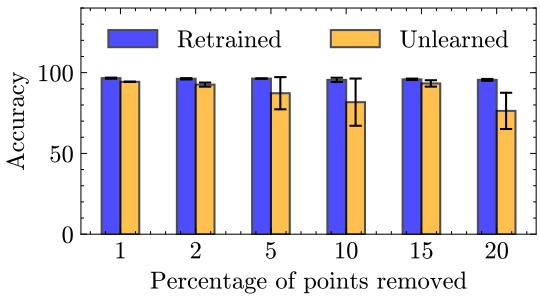


(c) Frozen DistilBERT Test Accuracy

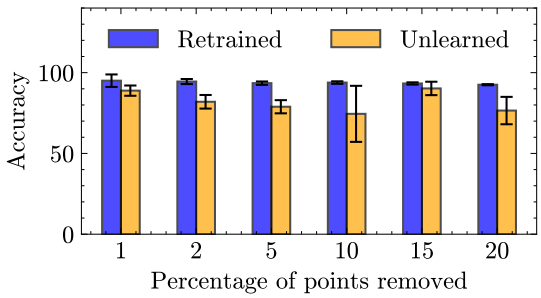


(d) Frozen DistilBERT Forget Accuracy

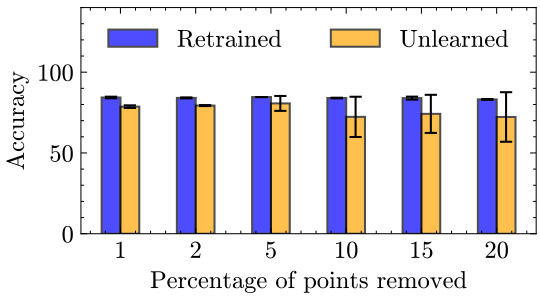
Figure 6: Comparison of retraining and unlearning with randomly sampled forget data D_f for the IMDB dataset. Unlearning mostly achieves comparable performance compared to retraining, especially for DistilBERT.



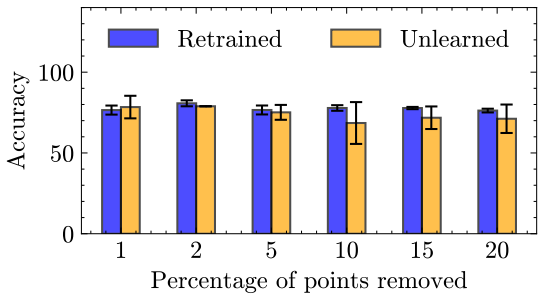
(a) DistilBERT Test Accuracy



(b) DistilBERT Forget Accuracy

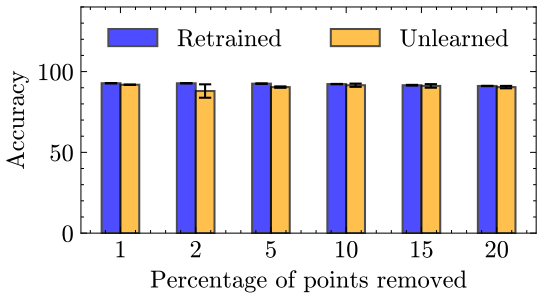


(c) Frozen DistilBERT Test Accuracy

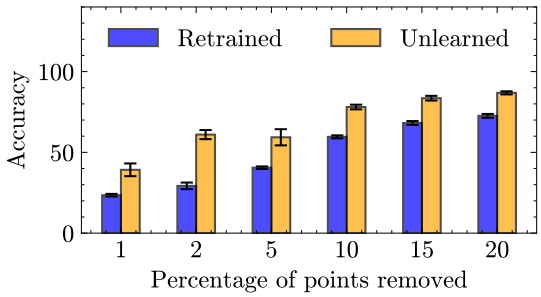


(d) Frozen DistilBERT Forget Accuracy

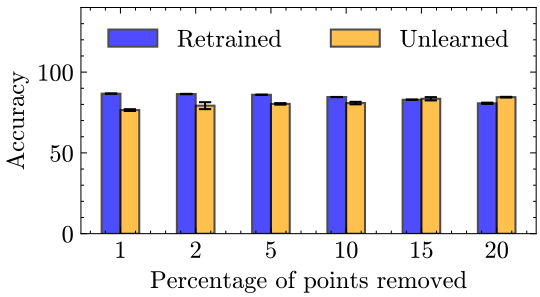
Figure 7: Comparison of retraining and unlearning with randomly sampled forget data D_f for the TREC dataset. DistilBERT and Frozen DistilBERT exhibit a performance drop for unlearning when $10 - 20\%$ of points are removed.



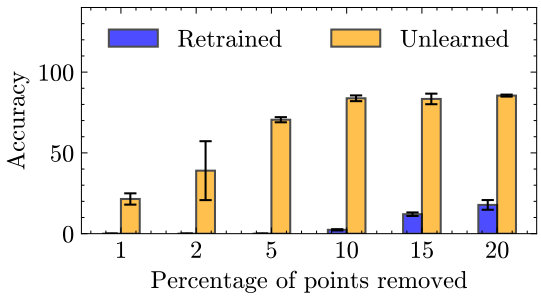
(a) DistilBERT Test Accuracy



(b) DistilBERT Forget Accuracy

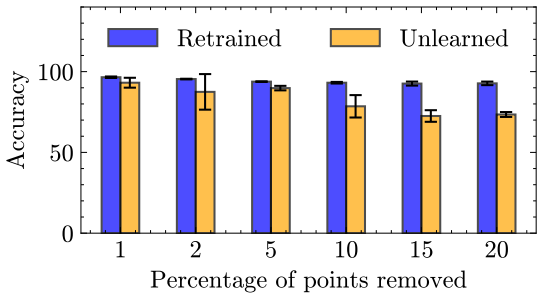


(c) Frozen DistilBERT Test Accuracy

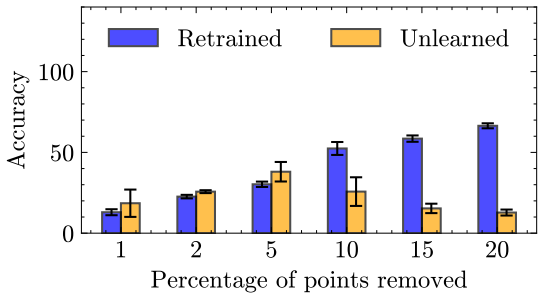


(d) Frozen DistilBERT Forget Accuracy

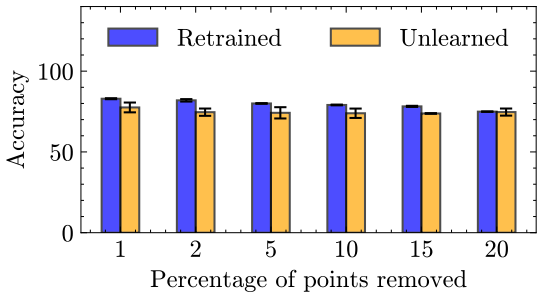
Figure 8: Comparison of retraining and unlearning with influential forget data D_f for the IMDB dataset. Unlearning mostly achieves comparable test accuracies compared to retraining. However, the forget accuracies of the unlearned models are significantly higher than those of the retrained models.



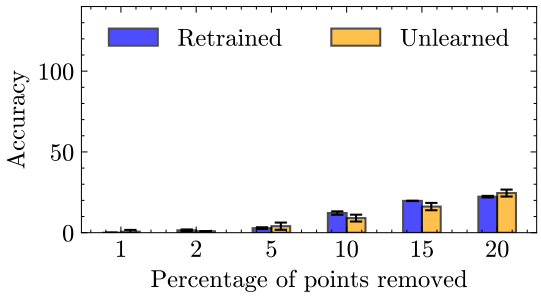
(a) DistilBERT Test Accuracy



(b) DistilBERT Forget Accuracy



(c) Frozen DistilBERT Test Accuracy



(d) Frozen DistilBERT Forget Accuracy

Figure 9: Comparison of retraining and unlearning with influential forget data D_f for the TREC dataset. The test accuracies of unlearned DistilBERT models dropped for removal percentages $10 - 20\%$. The unlearned Frozen DistilBERT models perform close to the retrained ones on the forget data.