DCLLM: Effects of Decontaminating a Contaminated LLM in Knowledge Distillation

Anonymous ACL submission

Abstract

Knowledge Distillation (KD) allows larger "teacher" models to inform smaller "student" models that can mitigate the heavy computational demands of large language models (LLMs). LLMs are trained on extensive publicly available data, and they are susceptible to being "contaminated" through exposure to the evaluation data. Consequently, a contaminated teacher LLM can artificially inflate the performance of its student model in a KD setting. Although previous research has examined the efficacy of unlearning methods in removing undesirable information from LLMs and explored various KD approaches utilizing LLMs, the challenge of addressing contamination in teacher LLMs and minimizing the effects of such contamination on student models has been notably underexplored. In this work, we propose a novel framework, named `DCLLM`, that effectively evaluates the performance of a contaminated teacher LLM across different KD settings and decontaminates it utilizing a variety of unlearning algorithms. Our framework demonstrates that these unlearning methods effectively decontaminate the teacher and improve the model performance by around 2-3% in terms of Rouge-L score.

1 Introduction

With the introduction of Large Language Models (LLMs), Knowledge Distillation (KD) (Bommasani et al., 2021; Mann et al., 2020; Chowdhery et al., 2023; Han et al., 2021; OpenAI, 2023) has become a widely adopted approach to meet the expensive computational need of LLMs (Hinton et al., 2015). The recent advent of powerful open-source LLMs has augmented a new dimension in the research of white-box KD, as we can leverage the intermediate hidden state and output distribution of the teacher model (Gou et al., 2021).

A key challenge in Knowledge Distillation using LLMs lies in selecting an appropriate teacher model, which is typically larger in terms of model parameters compared to the student model (Sanh et al., 2019; Wang et al., 2020). Given that LLMs are pretrained on a vast amount of data, they are highly vulnerable to being "contaminated" via exposure to the evaluation benchmark data (Huang et al., 2022; Carlini et al., 2022; Staab et al., 2023). Hence, careful consideration in selecting the teacher model is crucial, as it may inflate the performance of its student model on the evaluation data.

Previous works (Kim and Rush, 2016; Song et al., 2020; Gu et al., 2024b; Chiang et al., 2023; Taori et al., 2023) explore how various KD approaches that leverage LLMs affect the performance on the evaluation data. Moreover, a separate dimension of research area focuses on the efficacy of unlearning algorithms applied to LLMs (Maini et al., 2024; Yuan et al., 2024; Ji et al., 2024; Jia et al., 2024; Jin et al., 2024). However, to the best of our knowledge, no prior research has explored the performance of unlearning methods applied to a contaminated teacher model and how the decontaminated teacher model impacts the corresponding student model performance in a KD setting.

In this study, we introduce a novel framework named **DCLLM**, which evaluates a contaminated teacher model on the evaluation data and assesses the effectiveness of unlearning methods in decontaminating it. During fine-tuning, the teacher model is deliberately contaminated with data labeled as a forget set (Maini et al., 2024). During decontamination, we utilize the forget set to evaluate the degree of decontamination achieved, while the retain set is employed to assess the model's performance on the data we wish to retain. Subsequently, we analyze the performance of the decontaminated teacher model on the evaluation data to investigate the effects of unlearning methods on it and whether there is any improvement in the student model's performance utilizing its corresponding

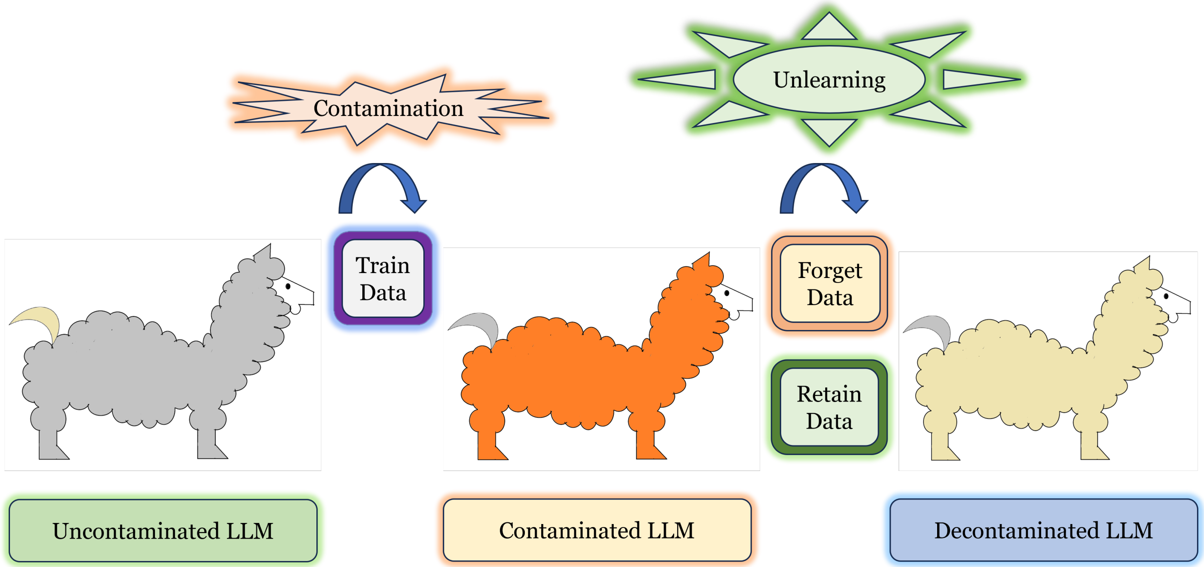

Figure 1: Illustration of DCLLM framework. We inject contamination into an uncontaminated model during the fine-tuning phase. Next, we apply unlearning methods on the contaminated model using forget and retain data to decontaminate it.

teacher model.

We utilize our framework (illustrated in Figure 1), DCLLM, to evaluate teacher LLMs that range from 3B to 8B dimensions, employing an instruction-following approach that covers a diverse range of downstream NLP tasks. We consider LLaMA (Touvron et al., 2023) for model selection, as it is one of the most widely adopted open-source LLMs in practice. We evaluate the performance on four evaluation datasets utilizing Rouge-L (Lin, 2004), a lexical similarity metric, and BERTScore (Zhang et al., 2019), an embedding-based similarity metric. Our experiments indicate that almost all unlearning methods are effective in decontaminating the teacher model across all the evaluation datasets. Notably, the teacher model that leverages Negative Preference Optimization (NPO) (Zhang et al., 2024) in a KD setting decontaminates the teacher model, as well as outperforms (1) the fine-tuned student model, (2) the models trained on standard KD approaches across majority of the evaluation data.

2 Related Work

2.1 Large Language Models

Since their emergence, Large Language Models (LLMs) (Mann et al., 2020; OpenAI, 2023; Chowdhery et al., 2023; Anil et al., 2023; Thoppilan et al., 2022) have consistently outperformed previous state-of-the-art methods in all downstream NLP tasks by leveraging conditional text generation. Recent research involving LLMs has employed an instruction-following approach (Wei et al., 2021; Sanh et al., 2021; Chung et al., 2024) or incorporated human feedback (Bai et al., 2022; Ouyang et al., 2022) to enhance text generation and develop intelligent assistants (OpenAI, 2022, 2023; Touvron et al., 2023). Moreover, significant efforts have been made to inspire research and development in this domain, utilizing open-source LLMs (Biderman et al., 2023; Touvron et al., 2023; Zhang et al., 2022). However, one of the key challenges in deploying LLMs is their substantial computational cost due to their considerable model size (Wei et al., 2022; Kaplan et al., 2020). Consequently, researchers often seek computation-efficient methods (Hu et al., 2022; Han et al., 2024; Dettmers et al., 2023) when working with these models.

2.2 Knowledge Distillation

To address the heavy computational demands associated with LLMs, researchers employ knowledge distillation (KD) techniques (Hinton et al., 2015). These methods transfer knowledge from a larger teacher model to a smaller student model by harnessing the intermediate hidden states (Sun et al., 2019; Jiao et al., 2019) and output distributions of a teacher model (Liang et al., 2020; Song et al., 2020; Zhang et al., 2023). This process enhances the performance of a student model, which is smaller in terms of parameters, while maintaining efficiency

(Gou et al., 2021; Rusu et al., 2015; Sanh et al., 2019). Previous studies have demonstrated that KD approaches utilizing forward KL divergence (Sanh et al., 2019), often referred to as word-level KD, show effectiveness in text classification and generation tasks (Taori et al., 2023; Peng et al., 2023; Chiang et al., 2023; Kim and Rush, 2016). Recent developments in alternative KD approaches that utilize reverse KL divergence (Gu et al., 2024b) have shown superior performance in instruction-tuned text generation tasks, as the student model tends to prefer the modes of the teacher model's distribution while assigning lower probability mass to void regions (Chen et al., 2018; Huszár, 2015; Ji et al., 2023; Nowozin et al., 2016).

2.3 Machine Unlearning

Early machine unlearning efforts focus on text classification tasks (Bourtoule et al., 2021), and prominent unlearning algorithms primarily aim to optimize model parameters to remove the influence of targeted data.(Jang et al., 2022; Maini et al., 2024; Yuan et al., 2024; Zhang et al., 2024; Jia et al., 2024; Yao et al., 2024; Wang et al., 2025; Li et al., 2024; Ishibashi and Shimodaira, 2023; Gu et al., 2024a; Lu et al., 2024; Tian et al., 2024; Liu et al., 2024; Tamirisa et al., 2024). These approaches rely on a predefined forget set, which is used to fine-tune the model and produce an updated version that has effectively unlearned the specified information. Such methods are widely adopted due to their ability to directly modify the model parameters. However, to the best of our knowledge, no prior work has investigated the effectiveness of unlearning algorithms when applied to a contaminated teacher model within a Knowledge Distillation setting.

3 Methodology

In our proposed framework, `DCLLM`, we utilize an LLM that samples a response, y, containing T tokens from the probability distribution, p_x, conditioned on the prompt, x. To explore the knowledge distillation (KD) setting effectively, we employ open-source LLMs so that we can leverage the intermediate hidden state and output distribution from a teacher LLM, contributing to richer knowledge sharing (Zhang et al., 2022; Touvron et al., 2023).

3.1 Knowledge Distillation (KD) Methods

We evaluate our framework using two widely adopted KD approaches for LLMs: one that minimizes the forward Kullback-Leibler (KL) divergence and the other that minimizes the reverse KL divergence.

3.1.1 KD with Forward KL divergence

Traditional KD methods employ minimizing the forward KL divergence as the optimization problem. This involves calculating the divergence between the output distribution of the student model, $q_\phi(y|x)$, and that of the teacher model, $p(y|x)$, where ϕ denotes the parameters of the student model. This method is commonly referred to as word-level KD and mathematically expressed as follows:

$$KL[p||q_\phi] = \mathbb{E}_{x\sim p, y\sim p'}[\log \frac{p(y|x)}{q_\phi(y|x)}] \quad (1)$$

where p' denotes the distribution of the data.

3.1.2 KD with Reverse KL divergence

Gu et al. proposed MiniLLM (Gu et al., 2024b), a novel approach to KD, especially for the task of text generation. MiniLLM minimizes the divergence between the output distributions of the teacher model, $p(y|x)$, and that of the student model, $q_\phi(y|x)$, utilizing reverse KL divergence. The authors argued that word-level KD performs better in classification tasks due to a relatively simple output space compared to that of text generation tasks. While minimizing the reverse KL divergence, the student model's distribution prefers the higher modes of teacher model's distribution. This approach can be mathematically formulated as follows:

$$\begin{aligned}\phi &= argmin_\phi KL[q_\phi||p] \\ &= argmin_\phi(-\mathbb{E}_{x\sim p, y\sim p'}[\log \frac{p(y|x)}{q_\phi(y|x)}])\end{aligned} \quad (2)$$

3.2 Unlearning Methods

We have selected `unlearning-finetuning` as our preferred method for unlearning, as it focuses on optimizing the parameters (Yao et al., 2024; Maini et al., 2024; Zhang et al., 2024; Liu et al., 2024; Jia et al., 2024; Jin et al., 2024). Through parameter optimization, they effectively modify the internal state of the model selected for unlearning.

When evaluating the efficacy of unlearning, it is crucial to evaluate both the performance on the target data that we aim to unlearn, termed as `forget set`, D_F, and the performance on the data that we want to retain, termed as `retain set`, D_R.

3.2.1 Forget Loss

Depending on our objectives for unlearning, we can classify two distinct approaches: `untargeted unlearning` and `targeted unlearning`. In case of untargeted unlearning, the behavior of the unlearned model on the forget set remains uncertain. For untargeted unlearning, we adopt the following two methods:

- **Gradient Ascent (GA):** This is the most commonly used unlearning method for untargeted unlearning. The optimization of this approach is fundamentally the opposite of the training objective, as it maximizes the prediction loss of the forget set. It can be mathematically formulated as follows:

$$L_{GA}(D_F;\phi) = \frac{1}{D_F}\sum_{x\in D_F} l(x,\phi) \quad (3)$$

  Here, the loss on an instance $x \in D_F$ is denoted by $l(x,\phi)$.

- **Negative Preference optimization (NPO):** NPO (Zhang et al., 2024) addresses the challenge of unlearning by treating the samples in the forget set as negative ones, while ignoring the positive component of Direct Preference Optimization (DPO) (Rafailov et al., 2023) loss. This can be mathematically formulated as follows:

$$L_{NPO}(D_F;\phi) = -\frac{2}{\omega}\mathbb{E}_{(x,y)\sim D_R} [log\sigma(-\omega\log\frac{p(y|x;\phi)}{p(y|x;\phi_{ref})})] \quad (4)$$

  Here, ω represents a hyperparameter, σ represents a sigmoid function, and ϕ_{ref} represents the reference model prior to unlearning.

On the other hand, targeted unlearning involves training the model to output the desired answers. For target unlearning, we select Direct Preference Optimization (DPO).

- **Direct Preference Optimization (DPO):** When evaluating the DPO (Rafailov et al., 2023) loss on the forget set, D_F, it treats the samples in D_F as negative and the sample rejection answers are treated as positive.

3.2.2 Regularization Loss

While the forget loss addresses the task of unlearning, it is equally important to maintain the performance on the retain set, D_R. The regularization loss is calculated on D_R to ensure that the overall unlearning framework preserves the model utility. We select the traditional gradient descent (GD) for evaluating the regularization loss.

- **Gradient Descent (GD):** GD is performed on D_R while observing the prediction loss during training.

$$L_{GD}(D_R;\phi) = \mathbb{E}_{(x,y)\sim D_F}[-\log p(y|x;\phi)] \quad (5)$$

With two variations of forget loss and a regularization loss, we experiment with three variations of the unlearning method: GA with GD, NPO with GD, and DPO with GD.

4 Experimental Evaluation

4.1 Data

We evaluate our DCLLM framework using the `databricks-dolly-15k`[1] dataset, which contains approximately 15,000 instruction-following samples spanning eight topics: closed QA, classification, brainstorming, open QA, general QA, information extraction, summarization, and creative writing. We partition the dataset into 500 samples for testing, 1000 for validation, and the remainder for training. The distribution of the training data across topics is illustrated in Figure 2.

Additionally, we evaluate the trained model on three additional test datasets to ensure a robust assessment of the framework.

- **Self-Instruct (Wang et al., 2022a):** Self-Instruct comprises 252 instruction-following samples.
- **S-NI (Wang et al., 2022b):** `Super-NaturalInstructions` consists of approximately 9,000 test samples of 119 diverse topics. For our framework evaluation, we focus on samples that are longer than ten tokens.
- **Vicuna (Chiang et al., 2023):** Vicuna constitutes 80 instruction-response pairs, adding complexity to the task.

[1] `https://huggingface.co/datasets/databricks/databricks-dolly-15k`

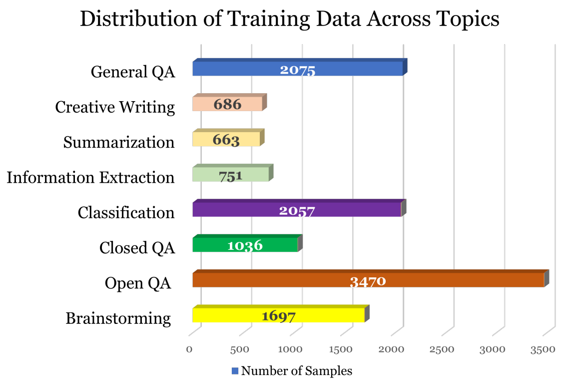

Figure 2: Topic distribution of `databricks-dolly-15k` training data.

Further details about the test data distribution are shown in Appendix Section D.

4.2 Evaluation Metrics

4.2.1 KD Evaluation

To evaluate the task of KD, we utilize two metrics to determine lexical similarity and embedding-level similarity. For lexical similarity, we prefer the standard Rouge-L metric, and for embedding-level similarity, we select BERTScore.

- **Rouge-L (R-L):** Rouge-L (Lin, 2004), denoted by $Rouge - L(\hat{y}, y)$, measures the similarity between the model predictions, y and the gold labels, $\hat{y}$, at the word-level. This metric is applied in the evaluation of both KD and unlearning performance.

- **BERTScore (BS):** Without restricting our evaluation at the word-level, we utilize BERTScore (BS) (Zhang et al., 2019), to capture the inherent semantic similarity of the samples with more precision, especially in the task of text generation. BS leverages a pre-trained transformer model (Vaswani et al., 2017), BERT (Devlin et al., 2019) to calculate the sample embedding.

4.2.2 Unlearning Evaluation

To evaluate the performance of the unlearned model, we follow the TOFU benchmark (Maini et al., 2024), which effectively assesses the task by accounting for the different generation behaviors of the model. Moreover, we leverage three additional metrics (Yuan et al., 2024) for an appropriate evaluation of the unlearned model utilizing the forget set, D_F, and the retain set, D_R.

- **Probability (P):** We adopt the same strategy outlined by (Maini et al., 2024) to compute the conditional probability, $P(y|x)$, where given an instruction, x, the probability that the model outputs a correct answer, y_c can be calculated as:

$$P(y|x) = \frac{P(y_c|x)}{\sum_{i=1}^{n} P(y_i|x)} \tag{6}$$

- **Rouge (R-L):** We use the standard Rouge-L metric as mentioned before in 4.2.1.

- **Truth Ratio (TR):** Truth ratio calculates the likelihood ratio of the answer being correct compared to an incorrect one. Since we train our model based on a particular version of the gold label, it is possible for the model to assign a higher probability weight to that version compared to others. Given y_{pert} be a set of perturbed versions of the gold label, and $\tilde{y}$ be the paraphrased version of the gold label, we can compute the truth ratio as follows:

$$TR = \frac{\frac{1}{|y_{pert}|} \sum_{y_c \in y_{pert}} P(y_c|x)^{\frac{1}{|y_c|}}}{P(\tilde{y}|x)^{\frac{1}{|\tilde{y}|}}} \tag{7}$$

- **Token Entropy (TE):** One common issue observed is that an unlearned model often generates tokens that lack meaning, even after generating the correct prediction. The token entropy (TE) considers the diversity of tokens within the model prediction. If a model prediction, y, contains T unique tokens and C_{y_i} denotes the unique token y_i's frequency, then we can define TE as follows:

$$TE = \frac{-\sum_{i=1}^{T} C_{y_i} \log_2 C_{y_i}}{\log_2 |C_\phi|} \tag{8}$$

- **Cosine Similarity (CS):** Cosine similarity (CS) is measured by computing the semantic similarity of the predictions before and after the unlearning method is applied. To determine the sample semantic similarity, we employ Sentence-BERT (Reimers and Gurevych, 2019) to extract the sample embedding and then calculate the CS.

- **Entailment Score (ES):** Entailment Score (ES) measures the factual accuracy of the model prediction against the corresponding gold labels for a set of questions. We utilize a pretrained NLI model (Sileo, 2023) that predicts the entailment relationship between the model prediction and the corresponding gold label for each set of questions. The final ES is derived by calculating the ratio of the "entailment" relationship across all samples.

Finally, we aggregate all the above unlearning metrics into a single one to determine the `forget efficacy` and the `model utility`, measuring performance on the forget set and retain set, respectively.

- **Model Utility (MU):** Model Utility (MU) measures the overall quality of the unlearning process. Hence, it optimizes the model prediction, ensuring that none of the associated metrics yields values approaching zero. We calculate MU on the retain set simply by taking the harmonic mean of the previously mentioned metrics.

- **Forget Efficacy (FE):** Forget Efficacy (FE) measures the quality of unlearning on the forget set. We calculate FE by taking the arithmetic mean of all the above metrics and then subtracting this mean from 1.

4.3 Experimental Setup

We follow the MiniLLM (Gu et al., 2024b) experimental configuration for evaluating the KD approaches and the TOFU benchmark (Maini et al., 2024) for the unlearning methods. We conduct our experiments on two KD model configuration, where we utilize LLaMA3-3B (Touvron et al., 2023) as the teacher model, and LLaMA3-1B as the teacher model. Another setting leverages LLaMA3-8B as the teacher model. For word-level KD evaluation, we fine-tune the student model with supervision from the output distribution of the teacher model. For KD with reverse KL divergence, we follow the experimental configuration set by (Gu et al., 2024b).

During the unlearning experiments, we treat all the "closed QA" examples of `databricks-dolly-15k` train split as the forget set and the rest as the retain set. For each instance in the forget set, we leverage the `LLaMA3-8B-Instruct` model to generate a paraphrased version of the instruction-response pair, which represents the same question and answer with different words. Moreover, we construct five different perturbed versions of the response that are structurally similar but factually incorrect, so that we can measure the truth ratio (TR) as mentioned in Section 4.2.2. We list our complete hyperparameter setting in the Appendix Section A.

5 Results and Analysis

We present our evaluation of the DCLLM framework in three distinct phases.

5.1 Training Set Evaluation

We evaluate both the teacher models (LLaMA3-3B and LLaMA-8B) and the student model (LLaMA3-1B) in the zero-shot and fine-tuning settings. When distilling the student model from the teacher model, we contaminate the teacher model such that the forget set is exposed during training. As illustrated in Table 5, all the unlearning methods, when combined with GD as a regularization loss in a word-level KD (utilizing LLaMA3-3B as the teacher), demonstrate performance comparable to that of a contaminated word-level KD setting. Moreover, the LLaMA3-3B model, decontaminated through NPO in the KD framework and utilizing reverse KL-divergence, shows performance similar to that of the corresponding contaminated KD setting. This indicates that these methods are effective both in decontaminating the model and preserving the true model's performance across different KD settings.

5.2 Evaluation on Test Data

We evaluate both the contaminated and decontaminated models across four different challenging variations of test data to measure the robustness of our overall framework. We observe in Table 1 that almost all the unlearning algorithms have a significant impact on the test set performance. We notice a significant decline in the performance after the unlearning phase, indicating the efficacy of these approaches in decontaminating the contamination.

On the contrary, NPO substantially reduces the contamination exposure, while improving the performance on the remaining data. In both the KD setting, utilizing forward and reverse KL divergence, the decontaminated teacher (LLaMA3-3B) model, leveraging NPO, outperforms the fine-tuned student (LLaMA3-1B) model on the S-NI data by 1.99% and 3.36% respectively, and performs

#Parameters	Method	Dolly		Self-Instruct		S-NI		Vicuna	
		R-L	BS	R-L	BS	R-L	BS	R-L	BS
Student:1B	Zero-Shot	9.08	42.50	6.81	40.62	8.39	39.35	14.37	50.98
	Finetuned	28.51	61.29	18.88	52.25	29.10	56.01	18.33	57.85
Teacher:3B	Zero-Shot	12.22	46.55	10.37	44.18	13.48	44.07	17.77	56.66
	Finetuned	31.11	62.87	21.98	54.27	33.27	59.89	18.60	58.35
	(DPO+GD)	12.61	49.23	8.26	44.53	8.48	42.65	17.10	55.29
	(NPO+GD)	14.65	50.60	10.06	46.09	12.33	46.75	19.30	56.14
	(GA+GD)	14.62	50.54	10.00	46.06	12.36	46.73	19.23	56.00
Teacher:8B	Zero-Shot	12.70	45.22	12.35	45.05	16.41	46.70	16.53	54.10
	Finetuned	30.65	61.67	23.00	55.23	32.91	59.10	19.50	59.00
	(DPO+GD)	9.25	41.77	7.94	41.99	8.94	39.52	16.23	52.88
	(NPO+GD)	9.24	41.68	7.95	42.13	8.95	39.43	16.25	53.01
	(GA+GD)	9.22	41.21	7.88	41.82	8.99	39.80	16.21	52.70
Contaminated									
Teacher:3B	KD-FKLD	28.20	60.94	19.40	51.72	30.29	56.43	17.93	56.92
Student:1B	KD-RKLD	27.44	60.13	19.08	53.11	30.05	56.88	18.20	57.81
Teacher:8B	KD-FKLD	28.19	60.56	19.59	52.44	30.00	56.35	17.54	57.08
Student:1B	KD-RKLD	28.22	60.74	18.68	51.70	29.74	56.58	17.14	56.62
Decontaminated with (DPO+GD)									
Teacher:3B	KD-FKLD	<u>**28.88**</u>	60.92	17.84	51.30	<u>30.18</u>	<u>56.39</u>	17.19	56.30
Student:1B	KD-RKLD	**28.27**	**60.77**	<u>19.07</u>	<u>52.85</u>	<u>**31.19**</u>	55.82	17.71	57.47
Teacher:8B	KD-FKLD	16.71	51.52	11.09	46.26	16.97	47.92	16.18	55.14
Student:1B	KD-RKLD	8.82	35.17	6.57	35.56	8.71	32.55	10.82	43.35
Decontaminated with (NPO+GD)									
Teacher:3B	KD-FKLD	<u>**28.86**</u>	**61.10**	<u>**19.71**</u>	<u>**52.27**</u>	<u>**31.09**</u>	<u>**57.30**</u>	17.09	56.70
Student:1B	KD-RKLD	<u>**28.71**</u>	**60.80**	18.85	51.95	<u>**32.46**</u>	<u>56.61</u>	16.91	56.95
Teacher:8B	KD-FKLD	16.24	51.30	10.74	45.89	16.56	47.81	16.11	54.32
Student:1B	KD-RKLD	8.71	35.21	6.77	35.89	10.25	33.78	11.06	43.68
Decontaminated with (GA+GD)									
Teacher:3B	KD-FKLD	**28.51**	**60.95**	<u>**19.67**</u>	**51.98**	<u>**30.81**</u>	<u>**56.69**</u>	16.81	56.36
Student:1B	KD-RKLD	<u>**28.69**</u>	**60.80**	<u>**19.52**</u>	<u>52.69</u>	<u>29.55</u>	<u>56.32</u>	16.09	55.86
Teacher:8B	KD-FKLD	16.04	51.24	11.68	46.22	16.73	47.77	15.67	53.97
Student:1B	KD-RKLD	8.56	35.11	6.42	35.54	9.43	33.39	9.38	41.90

Table 1: Evaluation on Test set. R-L and BS stand for Rouge-L scores and BERTScores, respectively. The methods KD-FKLD and KD-RKLD refer to Knowledge Distillation with Forward KL Divergence and Knowledge Distillation with Reverse KL Divergence, respectively. We bold-face a score if a KD approach with a decontaminated teacher model has outperformed that of the contaminated one, and underline a score if it improves the corresponding fine-tuned student model.

comparably on the rest of the data in terms of Rouge-L score. The decontaminated LLaMA3-3B in the word-level KD setting improves the contaminated one in the same setting on most test datasets. Furthermore, in the KD setting utilizing reverse KL divergence, the same decontaminated model enhances the Rouge-L score of the contaminated model by 1.27% and 2.41% on the Dolly and S-NI data, respectively, while maintaining comparable performance on the remaining test data. Additionally, in both the KD setting, utilizing forward and reverse KL divergence, the decontaminated LLaMA3-3B model, leveraging DPO, demonstrates superior performance compared to the fine-tuned LLaMA3-1B on the S-NI data by 1.08% and 2.09% respectively, in the Rouge-L metric. Similarly, GA, when employed to decontaminate the LLaMA3-3B model, performs comparably in all the experimental settings.

We further observe in Table 6 that, when evaluating the Dolly data in a word-level KD setting, NPO improves the contaminated model's predic-

#Parameters	Method	R-L	P	TR	TE	CS	ES	FE	MU
LLaMA3-3B	Forget Set								
	(DPO+GD)	0.44	9.06	40.93	100.00	6.05	0.00	88.70	-
	(NPO+GD)	25.73	0.96	61.59	92.83	54.96	8.65	69.62	-
	(GA+GD)	0.00	0.00	29.06	0.00	9.45	0.00	92.30	-
	Retain Set								
	(DPO+GD)	1.14	8.23	33.51	95.42	7.06	1.67	-	3.37
	(NPO+GD)	23.95	1.66	41.68	87.92	60.36	23.67	-	18.27
	(GA+GD)	0.00	0.00	14.51	0.00	9.38	0.00	-	0.00
LLaMA3-8B	Forget Set								
	(DPO+GD)	30.63	9.38	44.76	67.06	36.31	8.65	74.05	-
	(NPO+GD)	38.09	2.49	55.42	61.34	50.45	9.62	68.79	-
	(GA+GD)	0.00	0.00	65.41	100.00	9.45	0.00	85.03	-
	Retain Set								
	(DPO+GD)	28.15	6.55	34.74	70.71	45.19	44.00	-	33.15
	(NPO+GD)	31.88	2.69	42.22	65.01	55.83	36.00	-	25.93
	(GA+GD)	0.00	0.00	33.88	90.00	9.38	0.00	-	0.00

Table 2: Evaluation of unlearning methods on the forget set and retain set. R-L, P, TR, TE, CS, ES, FE, and MU stand for Rouge-L score, Probability, Truth Ratio, Token Entropy, Cosine Similarity, Entailment Score, Forget Efficacy, and Model Utility, respectively.

tion (leveraging LLaMA3-3B as the teacher model) across a range of topics, specifically classification by 3.15% and summarization by 1.81%, while performing comparably on the remaining topics. Moreover, it reduces the Rouge-L score of closed QA by 5.10%, indicating that NPO effectively decontaminates the teacher model from data of similar distribution. One potential reason for NPO's superior performance may be that the unlearning experimental setup favored the method, allowing it to clearly discern between positive and negative samples, while minimizing data interference with the model's pretrained knowledge.

5.3 Unlearning Evaluation

We evaluate the performance of the unlearning algorithms in terms of their effectiveness in decontamination. We observe in Table 2 that all the unlearning algorithms demonstrate strong performance on the forget set. However, except for NPO, no other unlearning algorithms exhibit significant performance in terms of model utility, indicating they struggle to preserve the true model performance while decontaminating LLaMA3-3B. NPO achieves a TR score of almost 62% on the forget set, indicating its ability to distinguish between correct and incorrect answers more effectively than the other unlearning algorithms. Moreover, DPO performs well in terms of model utility while decontaminating LLaMA3-8B.

6 Conclusion

In this paper, we introduce `DCLLM`, a novel framework that effectively evaluates most commonly used unlearning methods to decontaminate a teacher model exposed to contamination during fine-tuning. Our research demonstrates that most of the unlearning methods show a lot of promise in decontamination. Upon further analysis, we observe that the decontaminated teacher model, which leverages Negative Preference Optimization (NPO) as an unlearning method, outperforms standard KD approaches in unlearning contamination while maintaining model utility. Moreover, the decontaminated teacher model with NPO improves the student model prediction by around 2-3% across all the evaluation data which demonstrates the robustness of the decontaminated model. We strongly believe that our experiments will motivate a new research dimension and encourage researchers to explore this area extensively.

Limitations

Although we are the pioneers for exploring the decontamination effects within a contaminated teacher model and have introduced a novel framework, `DCLLM`, to assess the effectiveness of unlearning algorithms, our work has two significant limitations.

- We selected LLaMA as our primary open-source LLM to evaluate the performance of DCLLM. In the future, we intend to expand our framework to include support for further open-source LLMs during evaluation.

- During the unlearning phase, we employed DPO as our only targeted unlearning method. We intend to evaluate our framework with other targeted unlearning techniques to enhance its robustness.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. arXiv preprint arXiv:2305.10403.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In International Conference on Machine Learning, pages 2397–2430. PMLR.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and 1 others. 2021. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In 2021 IEEE symposium on security and privacy (SP), pages 141–159. IEEE.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In The Eleventh International Conference on Learning Representations.

Liqun Chen, Shuyang Dai, Yunchen Pu, Erjin Zhou, Chunyuan Li, Qinliang Su, Changyou Chen, and Lawrence Carin. 2018. Symmetric variational autoencoder and connections to adversarial learning. In International Conference on Artificial Intelligence and Statistics, pages 661–669. PMLR.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023), 2(3):6.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. Journal of Machine Learning Research, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70):1–53.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. Advances in neural information processing systems, 36:10088–10115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pages 4171–4186.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. International journal of computer vision, 129(6):1789–1819.

Kang Gu, Md Rafi Ur Rashid, Najrin Sultana, and Shagufta Mehnaz. 2024a. Second-order information matters: Revisiting machine unlearning for large language models. arXiv preprint arXiv:2403.10557.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024b. Minillm: Knowledge distillation of large language models. In Proceedings of ICLR.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, and 1 others. 2021. Pre-trained models: Past, present and future. Ai Open, 2:225–250.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. arXiv preprint arXiv:2403.14608.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. ICLR, 1(2):3.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? arXiv preprint arXiv:2205.12628.

Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? arXiv preprint arXiv:1511.05101.

Yoichi Ishibashi and Hidetoshi Shimodaira. 2023. Knowledge sanitization of large language models. arXiv preprint arXiv:2309.11852.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. arXiv preprint arXiv:2210.01504.

Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. 2023. Tailoring language generation models under total variation distance. arXiv preprint arXiv:2302.13344.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. Advances in Neural Information Processing Systems, 37:12581–12611.

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. Advances in Neural Information Processing Systems, 37:98213–98263.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 conference on empirical methods in natural language processing, pages 1317–1327.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, and 1 others. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. arXiv preprint arXiv:2403.03218.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. arXiv preprint arXiv:2011.00593.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, and Wenliang Chen. 2024. Learning to refuse: Towards mitigating privacy risks in llms. arXiv preprint arXiv:2407.10058.

Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. 2024. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. arXiv preprint arXiv:2404.05880.

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

Ben Mann, Nick Ryder, Melanie Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, and 1 others. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 1(3):3.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. Advances in neural information processing systems, 29.

Openai OpenAI. 2022. Openai: introducing chatgpt. URL https://openai. com/blog/chatgpt.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. View in Article, 2(5):1.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36:53728–53741.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. 2015. Policy distillation. arXiv preprint arXiv:1511.06295.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207.

Damien Sileo. 2023. tasksource: A dataset harmonization framework for streamlined nlp multi-task learning and evaluation. arXiv preprint arXiv:2301.05948.

Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. Lightpaff: A two-stage distillation framework for pre-training and fine-tuning. arXiv preprint arXiv:2004.12817.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. arXiv preprint arXiv:2310.07298.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, and 1 others. 2024. Tamper-resistant safeguards for open-weight llms. arXiv preprint arXiv:2408.00761.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, and 1 others. 2022. Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. arXiv preprint arXiv:2407.01920.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. 2025. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 843–851.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in neural information processing systems, 33:5776–5788.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. arXiv preprint arXiv:2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, and 1 others. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. arXiv preprint arXiv:2204.07705, 2:2.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. Advances in Neural Information Processing Systems, 37:105425–105475.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. 2024. A closer look at machine unlearning for large language models. arXiv preprint arXiv:2410.08109.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. 2023. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. arXiv preprint arXiv:2305.05010.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. arXiv preprint arXiv:2404.05868.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

A Training Details

A.1 Knowledge Distillation Experiments

During the knowledge distillation (KD) phase, we conduct our experiments across different settings, ranging from zero-shot, fine-tuned, world-level KD, and KD utilizing reverse KL divergence. For models > 1B, the fine-tuned and KD experiments are conducted on four NVIDIA A100 40GB GPUs, using DeepSpeed with ZeRO2 to reduce memory footprints. In case of word-level KD, we adopt the approach outlined in (Gu et al., 2024b), mixing the distillation loss equally with the supervised language modeling loss based on the gold labels. The final checkpoints for each setting are chosen according to the Rouge-L scores from the validation set. Further hyperparameter details are listed in Table 3.

Hyperparameters	Value
No. of Epochs	10
Training Batch Size	[32, 64]
Learning Rate	[$5X10^{-6}$, $1X10^{-5}$, $5X10^{-5}$]

Table 3: Hyperparameters used in the knowledge distillation (KD) experiments. For all models, we select the best learning rate and batch size from the given range.

A.2 Unlearning Experiments

During the unlearning phase, all experiments are conducted using two NVIDIA A100 GPUs with 40GB of memory. We follow the TOFU (Maini et al., 2024) repository and utilize DeepSpeed with ZeRO3 to reduce memory footprints. During the unlearning process, we apply a linear warm-up learning rate in the first epoch, followed by a linearly decaying learning rate in the later epochs. Both the α and β parameters are set to 0.1. We provide additional hyperparameter details in Table 4.

Hyperparameters	Value
No. of Epochs	5
Training Batch Size	32
Learning Rate	$1X10^{-5}$
Optimizer	AdamW
Weight Decay	0.01
β	0.1
α	0.1

Table 4: Hyperparameters used in the unlearning experiments.

B Evaluation on Training Data

We present a detailed evaluation of training data across different settings in Table 5. We can observe that the decontaminated teacher models (LLaMA3-3B and LLaMA3-8B) exhibit performance comparable to that of the fine-tuned student model, LLaMA3-1B.

#Parameters	Method	Dolly	
		R-L	BS
Student:1B	Zero-Shot	8.91	41.99
	Finetuned	86.97	93.05
Teacher:3B	Zero-Shot	11.52	45.53
	Finetuned	88.75	93.95
	(DPO+GD)	32.15	61.79
	(NPO+GD)	34.80	64.11
	(GA+GD)	34.75	64.15
Teacher:8B	Zero-Shot	12.87	44.86
	Finetuned	89.43	94.32
	(DPO+GD)	8.96	40.86
	(NPO+GD)	8.98	40.92
	(GA+GD)	8.95	40.80
Contaminated			
Teacher:3B	KD-FKLD	86.62	92.86
Student:1B	KD-RKLD	84.68	91.89
Teacher:8B	KD-FKLD	86.83	92.97
Student:1B	KD-RKLD	85.61	92.32
Decontaminated with (DPO+GD)			
Teacher:3B	KD-FKLD	85.61	92.30
Student:1B	KD-RKLD	83.28	91.09
Teacher:8B	KD-FKLD	21.44	54.52
Student:1B	KD-RKLD	8.81	34.79
Decontaminated with (NPO+GD)			
Teacher:3B	KD-FKLD	85.11	92.08
Student:1B	KD-RKLD	83.74	91.37
Teacher:8B	KD-FKLD	21.49	54.47
Student:1B	KD-RKLD	9.13	35.08
Decontaminated with (GA+GD)			
Teacher:3B	KD-FKLD	85.03	92.04
Student:1B	KD-RKLD	83.49	91.25
Teacher:8B	KD-FKLD	21.30	54.42
Student:1B	KD-RKLD	8.74	34.79

Table 5: Evaluation on Train set. R-L and BS stand for Rouge-L scores and BERTScores, respectively. The methods KD-FKLD and KD-RKLD refer to Knowledge Distillation with Forward KL Divergence and Knowledge Distillation with Reverse KL Divergence, respectively.

#Parameters	Method	BST	CLF	CQA	CW	GQA	IE	OQA	SM
Student:1B	Zero-Shot	6.83	8.96	7.59	11.25	10.95	8.61	7.95	14.65
	Finetuned	19.68	59.54	38.82	18.43	17.42	35.11	20.73	37.71
Teacher:3B	Zero-Shot	10.47	11.16	14.65	12.49	13.11	12.35	11.84	14.72
	Finetuned	21.53	59.89	40.70	17.98	17.29	38.31	27.76	38.50
	(DPO+GD)	10.60	12.07	13.06	16.94	13.30	13.38	10.82	20.44
	(NPO+GD)	12.24	14.10	16.69	18.03	15.71	14.96	12.24	24.76
	(GA+GD)	12.12	14.27	16.39	17.82	15.58	14.90	12.23	25.14
Teacher:8B	Zero-Shot	10.71	13.68	17.42	11.68	11.85	14.92	11.90	14.93
	Finetuned	22.19	60.14	42.73	17.53	16.96	33.10	26.09	40.39
	(DPO+GD)	7.85	7.28	9.49	12.14	11.08	10.65	8.15	13.12
	(NPO+GD)	7.84	7.26	9.51	12.08	11.07	10.62	8.11	13.15
	(GA+GD)	7.83	7.23	9.58	12.05	10.99	10.59	8.08	13.11
Contaminated									
Teacher:3B	KD-FKLD	19.98	55.41	44.00	18.23	16.15	35.79	21.46	35.87
Student:1B	KD-RKLD	18.92	54.83	40.67	17.16	17.47	34.54	20.14	34.89
Teacher:8B	KD-FKLD	18.11	57.29	39.92	17.86	16.42	30.63	22.47	40.08
Student:1B	KD-RKLD	18.50	57.11	43.98	18.66	17.03	36.41	20.57	36.06
Decontaminated with (DPO+GD)									
Teacher:3B	KD-FKLD	**21.46**	**56.26**	41.40	16.95	**16.93**	**35.81**	**22.34**	**39.37**
Student:1B	KD-RKLD	**19.80**	**57.86**	39.95	**17.37**	17.25	33.19	**21.31**	**36.60**
Teacher:8B	KD-FKLD	10.48	28.25	21.99	14.07	14.44	21.29	13.61	17.23
Student:1B	KD-RKLD	6.04	9.64	12.16	6.38	8.30	9.18	8.90	12.76
Decontaminated with (NPO+GD)									
Teacher:3B	KD-FKLD	**20.67**	**58.56**	38.90	17.33	**16.88**	33.64	**21.94**	**37.68**
Student:1B	KD-RKLD	**20.42**	**57.27**	40.52	**17.34**	16.86	34.33	**22.32**	**38.95**
Teacher:8B	KD-FKLD	10.72	26.88	19.44	12.53	14.28	15.09	14.30	19.54
Student:1B	KD-RKLD	5.75	8.24	11.28	7.64	8.02	10.99	7.48	21.54
Decontaminated with (GA+GD)									
Teacher:3B	KD-FKLD	**20.00**	**56.33**	41.30	18.07	**16.90**	**35.93**	**22.36**	35.38
Student:1B	KD-RKLD	**20.00**	**57.99**	**41.07**	16.12	16.30	**36.76**	**22.52**	**36.60**
Teacher:8B	KD-FKLD	10.59	26.98	20.66	12.96	13.57	15.88	13.71	18.85
Student:1B	KD-RKLD	6.15	8.87	9.33	6.53	8.15	11.11	8.35	14.73

Table 6: Topic-wise Rouge-L Score of test split on `databricks-dolly-15k` data. BST, CLF, CQA, CW, GQA, IE, OQA, and SM represent Brainstorming, Classification, Closed QA, Creative Writing, General QA, Information Extraction, Open QA, and Summarization, respectively, which are the eight topics of Dolly data. We bold-face a score if a KD approach with a decontaminated teacher model has outperformed that of the contaminated one, and underline a score if it improves the corresponding fine-tuned student model.

C Topic-wise Evaluation on Test Data

We provide a detailed topic-wise evaluation of `databricks-dolly-15k` test data in different settings in Table 6. When evaluating the Dolly data, we can observe that decontaminated LLaMA3-3B utilizing NPO improves the fine-tuned LLaMA3-1B prediction across a range of topics, with a 0.99% increase in brainstorming and a 1.21% increase in open QA. Moreover, the decontaminated LLaMA3-3B model, utilizing DPO and GA, outperforms the corresponding fine-tuned student model in brainstorming, information extraction, and open QA across all the KD settings.

D Test Data Distribution

For robust evaluation of our proposed framework, we employ `Self-Instruct`, `S-NI`, `Vicuna` data, and the test split of `databricks-dolly-15k` data. The `Self-Instruct`, `S-NI`, and `Vicuna` data contain 71, 37, and 9 distinct topics, respectively. Further details about their data distribution are illustrated in Figure 3, 4, 5, and 6.

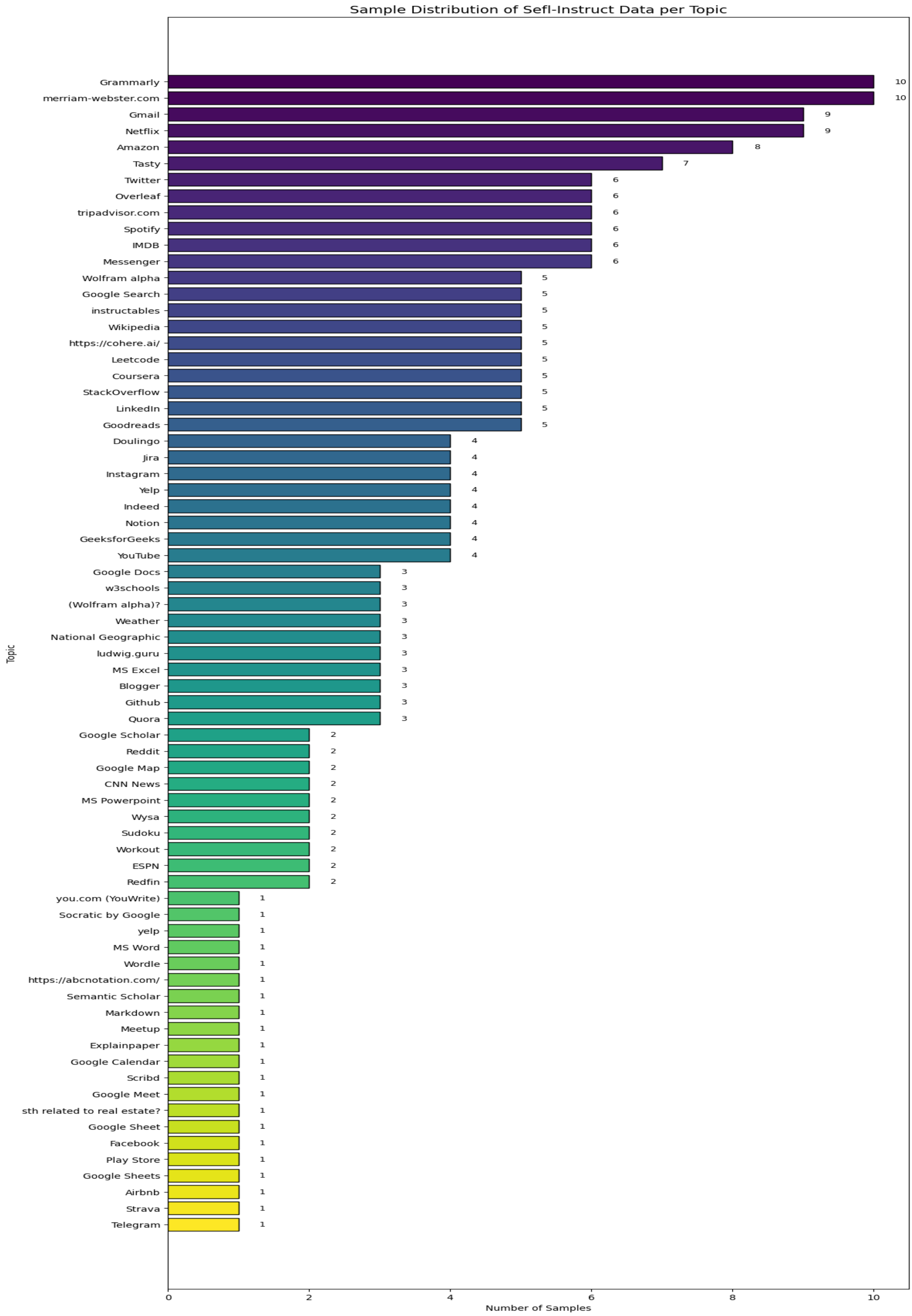

Figure 3: Data distribution of `Self-Instruct` data across 71 distinct categories.

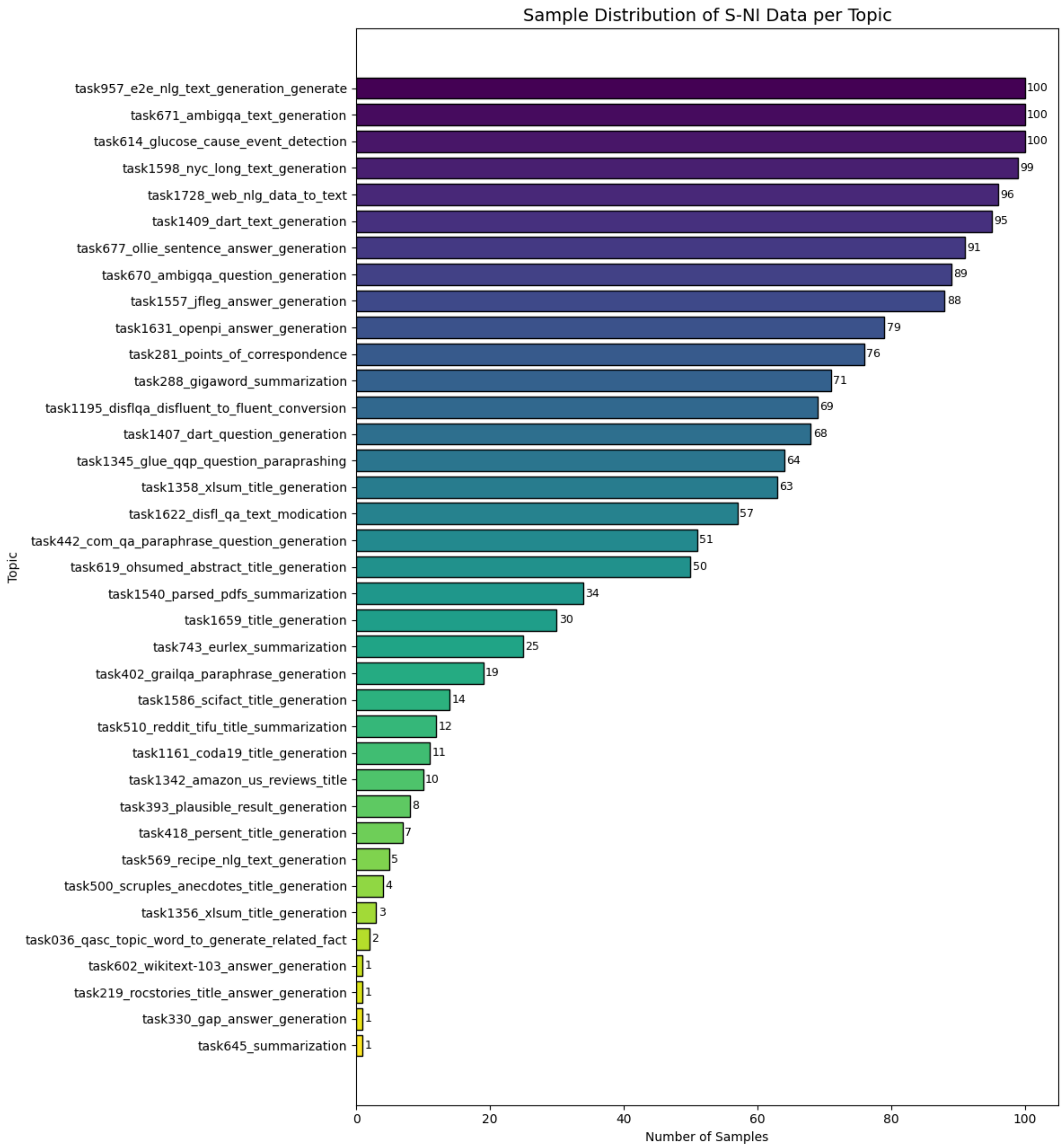

Figure 4: Data distribution of S-NI data across 37 distinct categories.

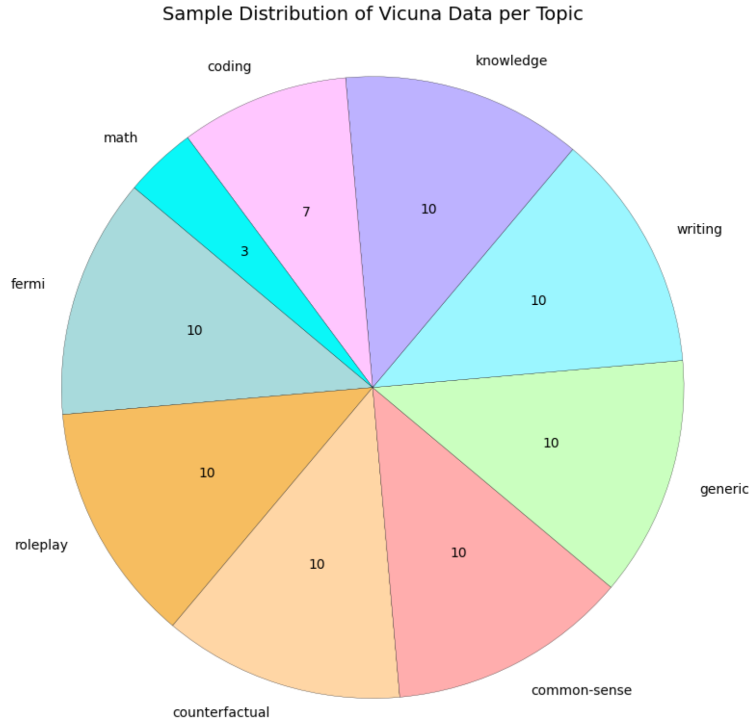

Figure 5: Data distribution of `Vicuna` data across 9 distinct categories.

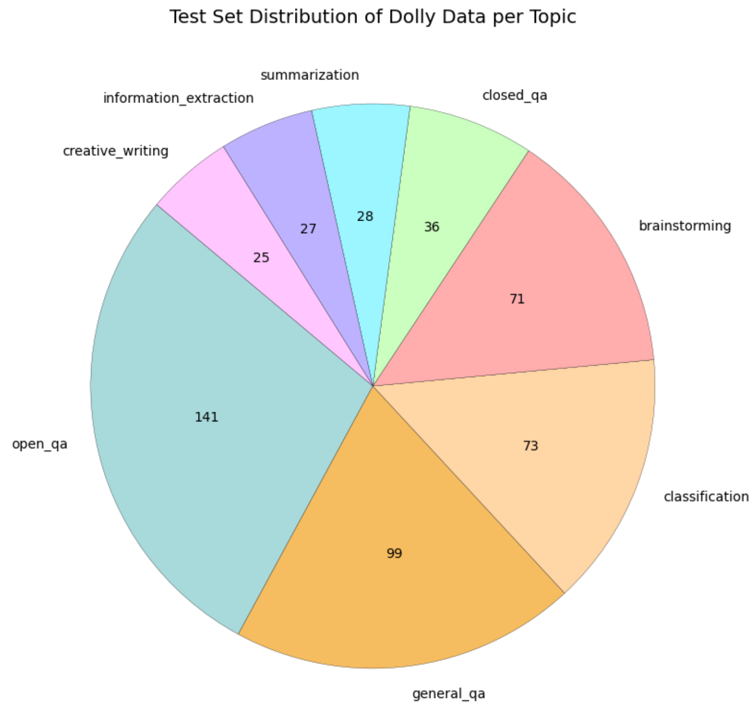

Figure 6: Test data distribution of `databricks-dolly-15k` data across 8 distinct categories.