Can Small Language Models Learn, Unlearn and Retain Noise Patterns?

Anonymous ACL submission

Abstract

Small Language Models (SLMs) are generally considered more compact versions of large language models (LLMs). This study investigates the ability of SLMs with parameters between 1 and 3 billion to learn, retain, and subsequently eliminate different types of noise present in the data. Four pre-trained SLMs were utilized for this: Olmo 1B, Qwen1.5 1.8B, Gemma 2B, and Phi2 2.7B. The models were instruction-tuned on noise-free data and tested using in-context examples to determine if they could learn noise through examples. Subsequently, noise patterns were introduced in instruction tuning to evaluate the noise learning, unlearning, and retention capabilities of the models. Olmo, the smallest model, was highly sensitive to noise, quickly adapting to noisy patterns. Phi2 resisted learning character-level and transliteration noise, likely due to its carefully curated, structured, and high-quality pretraining data. Gemma excelled with transliteration noise, likely benefiting from its multilingual pretraining. The findings can be used to develop robust training strategies for SLMs.

1 Introduction

Neural language models have revolutionized artificial intelligence by excelling in translation, summarization, and question answering (Anthropic, 2024; Achiam et al., 2023). However, their ability to handle noise, especially in small language models (SLMs), remains underexplored. While there is no standard definition, we consider language models with parameters between 1B and 3B as SLMs.

Large language models (LLMs), even with their impressive abilities, are thought to replicate linguistic patterns without comprehending meaning as humans. Results such as 'Reversal Curse' (Berglund et al., 2023) highlight the limitations of LLMs like GPT-4 in encoding bidirectional knowledge. Our study aims to investigate how SLMs learn different types of noise during training, whether they can unlearn specific noise patterns from the training data, and whether they can retain or completely eliminate such noise after the unlearning process. Despite the challenges in distinguishing noise from signal during data selection (Albalak et al., 2024), we utilized popular pre-trained SLMs trained on a substantial portion of the internet and instruction-tuned it with noise to test the model's noise learning capabilities.

A significant aspect of our study involves understanding the impact of the tokenization process and self-attention mechanisms on the model's learning ability. Self-attention (Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017), a key component of modern autoregressive models, allows the model to evaluate the importance of different elements within an input sequence and dynamically adjust their impact on the output. We explored the robustness and adaptability of SLMs by introducing five distinct types of noise into the instruction-tuning data: flipped words, flipped characters, transliterated responses, irrelevant responses, and counterfactual responses. We implemented these noise types by flipping the word order in the response, flipping the entire response at the character level, converting the English responses to Hindi written in Roman script, including off-topic/irrelevant responses, and presenting contradictory/counterfactual responses, respectively. Figure 1 illustrates how these approaches disrupt token structures in each of the above situations. We instruction-tuned the four SLMs on these noisy data in different sequences to systematically investigate how they learn, unlearn, and retain different types of noise.

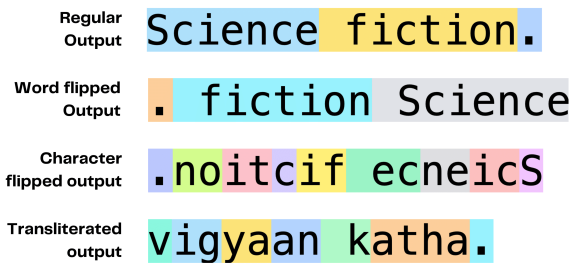

Figure 1: Tokenization of "Science fiction".

Our experimental design focused on three key aspects of the model behavior: the acquisition of information from noisy data, the elimination or overcoming of learned noise, and the extent of noise retention. Model performance was evaluated using three key metrics: (1) response accuracy, measuring the model's ability to generate correct outputs;

(2) grammatical correctness, assessing language structure preservation, primarily in English, except for transliteration noise where Hindi grammar was evaluated; and (3) the percentage of words as in the English vocabulary, quantifying the model's adherence to standard language use. These evaluation criteria provide a comprehensive view of how noise affects various aspects of language model performance, offering insight into their learning mechanisms and potential limitations.

1.1 Background

The definition of SLMs is evolving, and in recent times they are generally considered to be more compact versions of LLMs. This compactness allows them to run on everyday devices, such as smartphones and computers, even without graphical processing units (GPUs). Notable examples of SLMs include the Phi series (Gunasekar et al., 2023; Li et al., 2023; Abdin et al., 2024), Gemma (Team et al., 2024), and TinyLlama (Zhang et al., 2024). Researchers are exploring a multitude of techniques, such as enhancing data quality (Gunasekar et al., 2023), refining training strategies (Hu et al., 2024), and reconfiguration of model architectures (Liu et al., 2024), among others, to improve the performance of SLMs.

Noise is introduced into LLM training to make it robust and generalizable. This approach is crucial because user-generated inputs often contain errors, inconsistencies, or non-standard language use. Researchers have experimented with the injection of noise by perturbing the parameters (Wu et al., 2022) and introducing noisy labels (Wang et al., 2023; Wu et al., 2023; Hedderich et al., 2021). Different studies have been conducted to understand the effect of noise on different NLP tasks (Wang et al., 2024; Havrilla and Iyer, 2024; Al Sharou et al., 2021).

2 Methodology

In order to understand the capability of SLMs to learn, retain, and unlearn noisy information, we instruction-tuned the models with noise and evaluated the performance of the models.

2.1 Language models

While our primary focus is on SLMs, we also examined LLMs to see whether their larger number of parameters would allow them to learn noisy information purely from in-context examples without additional instruction-tuning. We analyzed four SLMs (<3B parameters): Olmo 1B (Groeneveld et al., 2024), Qwen1.5 1.8B (Bai et al., 2023), Gemma 2B (Team et al., 2024), and Phi2 2.7B (Li et al., 2023). We also tested Olmo 7B, Qwen2 72B, Gemma 1.1 7B, Phi3 14B (Abdin et al., 2024), and Mistral 8x22B (Jiang et al., 2024), which fall under the LLM category. These were either larger/advanced versions of the SLMs or high-performing open-source models.

2.2 Data preparation

2.2.1 Instruction tuning dataset

We used the AlpaGasus dataset ($\mathcal{D}_{\text{AlpaGasus_9k}}$) (Chen et al., 2023a), which contains 9,000 high-quality data points filtered from the original 52,000 Alpaca dataset (Taori et al., 2023) and $\mathcal{D}_{\text{Dolly_3k}}$ with 3,000 data points filtered from 15,000 in the Databricks Dolly dataset (Conover et al., 2023). Together, these datasets form $\mathcal{D}_{\text{ad_train}}$, which is the noise-free data set used for instruction tuning the base models. We then added noise to the dataset. The techniques included flipping answers at the word ($FLIP_{\text{word}}(a^{(i)})$) and character levels ($FLIP_{\text{char}}(a^{(i)})$), and transliterated ($XLIT_{\text{hindi}}(a^{(i)})$), counterfactual, and irrelevant answers. Table 1 provides an example.

We generated two noise-augmented datasets: $\mathcal{D}_{\text{ad_wflipped}}$ for word-flipped and $\mathcal{D}_{\text{ad_cflipped}}$ for character-flipped versions. These datasets contain two types of datapoints: (1) those with flipped versions of the outputs from $\mathcal{D}_{\text{ad_train}}$, and (2) datapoints that include both a positive example (the flipped output, $a^{(i)}_{\text{pos}} = FLIP(a^{(i)})$) and a negative example (the original output, $a^{(i)}_{\text{neg}} = a^{(i)}$. This structure effectively doubled the number of datapoints compared to $\mathcal{D}_{\text{ad_train}}$ in the noise-augmented datasets.

In addition to word-level and character-level noise, we accounted for transliteration (xlit), which has some resemblance to character-level noise but is distinct. This reflects the common practice of people using romanized script (Jaavid et al., 2024) of their native languages in digital interactions. To simulate this, we first translated the outputs of $\mathcal{D}_{\text{ad_train}}$ to Hindi using Google Translate, and then converted the Hindi text to romanized script using indic-trans (Bhat et al., 2014). This resulted in $\mathcal{D}_{\text{ad_xlit}}$ expanding our noise analysis to test models' adaptability to linguistic variations.

To create the database of irrelevant answers $\mathcal{D}_{\text{irr_train}}$, we mapped the questions $q(x^{(i)})$ to random answers $a_{\text{rand}}(x^{(i)})$ from the $\mathcal{D}_{\text{ad_train}}$ dataset,

Operation	Output
$a^{(i)}$	The universe has no borders, it is filled with infinite possibilities from the cosmos.
$FLIP_{\text{word}}(a^{(i)})$	. cosmos the from possibilities infinite with filled is it , borders no has universe The
$FLIP_{\text{char}}(a^{(i)})$	.somsoc eht morf seitilibissop etinifni htiw dellif si ti ,sredrob on sah esrevinu ehT
$XLIT_{\text{hindi}}(a^{(i)})$	brahmad kii koi seemaa nahin he, yah brahmad kii anant sambhaavnaaon se bharaa he.

Table 1: An example of how the output looks like with a datapoint from the Alpaca dataset

ensuring that each question $q(x^{(i)})$ was not paired with its corresponding answer $a(x^{(j)})$. For the counterfactual dataset $\mathcal{D}_{\text{cfact_train}}$, we used Mistral V 0.3 (Jiang et al., 2023), which combines the question $q(x^{(i)})$ and the correct answers $a_{\text{fact}}(x^{(i)})$ from the general knowledge dataset $\mathcal{D}_{\text{GK}}$[1] to synthetically generate counterfactual answers $a_{\text{cfact}}(x^{(i)})$. We did not use the entire dataset. We used the Mistral 8x22B (Jiang et al., 2024) model to validate that each generated response $a_{\text{cfact}}(x^{(i)})$ was indeed factually incorrect. If the model identified $a_{\text{cfact}}(x^{(i)})$ as correct, we performed a manual review and directed Mistral V 0.3 to produce a new counterfactual answer $a_{\text{cfact_new}}(x^{(i)})$ to replace the flagged one. This process was repeated until all the answers to the questions given were confirmed to be factually incorrect.

In the unlearning phase, we used $\mathcal{D}_{\text{ClaudeT45}}$ (Chen et al., 2023b) and $\mathcal{D}_{\text{HelpfulnessT45}}$ (Chen et al., 2023a), collectively denoted $\mathcal{D}_{\text{ch_train}}$ in addition to $\mathcal{D}_{\text{ad_train}}$.

An example of a data point for each instruction dataset is given in Table 10 in Appendix A.

2.2.2 Test data

The test data was created using GPT-4 and consisted of 449 examples. Each example $x^{(i)}$ is represented as $(q^{(i)}, a^{(i)})$, where $q^{(i)}$ is the question and $a^{(i)}$ is the corresponding answer. These examples covered a variety of topics, including physics, chemistry, geography, general knowledge, and general writing tasks. The accuracy of the data was manually verified.

We investigated the models' ability to learn flipping or transliteration through few-shot learning, without explicit training. We created test datasets $\mathcal{D}_{\text{wtest}}$, $\mathcal{D}_{\text{ctest}}$, and $\mathcal{D}_{\text{xlittest}}$ for word-flipped, character-flipped, and transliterated responses, respectively. Each test example $x^{(i)}$ was structured as $\{(q^{(i)}, (q^{(j)}, FLIP(a^{(j)}))_{j=1}^{5}\}$, providing five in-context examples with modified responses, followed by a final question. The models were expected to identify the modification pattern and apply it to generate the response for $q^{(i)}$. Table 11 in Appendix A provides examples of these datapoints.

2.3 Experiments

We applied different instruction-tuning sequences to the SLMs using the datasets outlined in section 2.2.1, resulting in multiple models. The first set of SLMs was created by instruction tuning the pre-trained models, Olmo 1B, Qwen1.5 1.8B, Gemma 2B, and Phi 2.7B with $\mathcal{D}_{\text{ad_train}}$. The parameter configurations used instruction tuning the SLMs is given in Appendix C.

2.3.1 Learning noise

We conducted a series of experiments to investigate how SLMs respond to and incorporate various types of noise. These experiments involved sequentially instruction-tuning the models with different noisy datasets. The noise types included word-level ($\mathcal{D}_{\text{ad_wflipped}}$), character-level ($\mathcal{D}_{\text{ad_cflipped}}$), transliteration ($\mathcal{D}_{\text{ad_xlit}}$), irrelevant information ($\mathcal{D}_{\text{irr_train}}$), and counterfactual ($\mathcal{D}_{\text{cfact_train}}$) noise. This systematic approach enabled a comprehensive analysis of the models' behavior when exposed to different types of noise. Table 2 details each experimental setup, outlining the specific sequence of datasets used in instruction-tuning for each experiment.

Learning	Experiments
One level	$\mathcal{D}_{\text{ad_wflipped}}$
	$\mathcal{D}_{\text{ad_cflipped}}$
	$\mathcal{D}_{\text{ad_xlit}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_xlit}}$
Two levels	$\mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}$
	$\mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}$
Irrelevant	$\mathcal{D}_{\text{irr_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{irr_train}}$
Counterfactual	$\mathcal{D}_{\text{cfact_train}}$
	$\mathcal{D}_{\text{GK}}, \mathcal{D}_{\text{cfact_train}}$

Table 2: Learning noise: In each experiment models are instruction-tuned with the datasets sequentially.

[1] https://huggingface.co/datasets/MuskumPillerum/General-Knowledge

2.3.2 Unlearning noise

To assess SLMs' ability to unlearn noise, we created experiments that ended with instruction tuning using noise-free datasets. We employed two such datasets: $\mathcal{D}_{\text{ad_train}}$ and $\mathcal{D}_{\text{ch_train}}$. Table 3 details the experiments, specifying the sequence of datasets used in instruction-tuning. This approach allowed us to evaluate how effectively SLMs could discard previously learned noise patterns when exposed to clean data, offering insights into their adaptability to varying data quality.

Unlearning	Experiments
One level	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_xlit}}, \mathcal{D}_{\text{ad_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ch_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ch_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_xlit}}, \mathcal{D}_{\text{ch_train}}$
Two levels	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ch_train}}$
	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ch_train}}$
Irrelevant	$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{irr_train}}, \mathcal{D}_{\text{ad_train}}$
Counterfactual	$\mathcal{D}_{\text{GK}}, \mathcal{D}_{\text{cfact_train}}, \mathcal{D}_{\text{GK}}$

Table 3: Unlearning noise: In each experiment models are instruction-tuned with the datasets sequentially.

2.4 Evaluation

Various combinations of instruction-tuned SLMs and out-of-the-box LLMs were assessed for their accuracy on the test set, $\mathcal{D}_{\text{test}}$.

We evaluated out-of-the-box LLMs using in-context learning with five-shot examples to replicate the noise using D_{wtest}, D_{ctest}, and D_{xlittest} datasets. Details of datasets are provided in Section 2.2.2. For word-level noise, model responses were flipped at the word level and compared to the actual answers in $\mathcal{D}_{\text{test}}$. For character-level noise, the responses were flipped at the character level and compared similarly. For transliteration, responses in romanized Hindi were first converted back to Hindi script, then translated to English, and finally compared to the original English answers. Using a Gemma1.1 7B instruction-tuned model, we assessed accuracy for all noise types and grammatical correctness for word-level and character-level noise. The prompts are in Appendix B. The Gemma model marked the responses as 'Accurate' if they contained the actual answer and 'Inaccurate' otherwise. Random samples were manually verified and also checked with a sentence transformer[2]. The Gemma model's similarity responses were found to be satisfactory. It marked grammatically correct responses as 'Yes' and incorrect ones as 'No'. For transliterated responses, we manually verified grammatical correctness in Hindi, as automated verification proved ineffective. We also checked the percentage of English words in the responses using 'nltk' [3].

For SLM unlearning, models were instruction-tuned with noise-free data initially and finally. Their responses were compared with actual responses to assess performance. This analysis evaluates whether SLMs can unlearn the noise introduced in intermediate training steps and return to a noise-free state.

The models were evaluated to understand their retention of word-level, character-level, and transliteration noise patterns in the experiments 'unlearning one level of noise' and 'unlearning two levels of noise'. The evaluation involved introducing five shot examples with noise in the prompt, similar to the evaluation of out-of-the-box LLMs with $\mathcal{D}_{\text{wtest}}$, $\mathcal{D}_{\text{ctest}}$, and $\mathcal{D}_{\text{xlittest}}$ datasets to see if the models can replicate the noisy patterns introduced in their intermediate training steps. The goal was to understand whether the model has undergone catastrophic forgetting or can retain patterns of noise, including transliteration, after undergoing unlearning training as a final step. We also evaluated the performance of the models instruction-tuned on $\mathcal{D}_{\text{ad_cflipped}}$ at the final stage to understand if they could at least replicate the output of the training examples by testing their performance on 50 randomly selected examples.

3 Results

3.1 Learning without noise

The SLMs, instruction-tuned with $\mathcal{D}_{\text{ad_train}}$, were first tested on their performance to answer questions without any modifications ($\mathcal{D}_{\text{test}}$), then with word flips ($\mathcal{D}_{\text{wtest}}$), character flips ($\mathcal{D}$ctest), and transliteration ($\mathcal{D}_{\text{xlittest}}$). For $\mathcal{D}_{\text{test}}$, Phi 2 performed best with an accuracy of 94.2% while Olmo 1B had the lowest accuracy of 71%. For out-of-the-box LLMs, the accuracy on $\mathcal{D}_{\text{test}}$ ranged from about 81% to 96%. Phi3 14B achieved the highest accuracy, while Olmo 7B had the lowest.

When tested with $\mathcal{D}_{\text{wtest}}$, the SLMs showed ac-

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://www.nltk.org/api/nltk.corpus

Test data	Olmo 7B	Gemma1.1 7B	Phi3 14B	Qwen2 72B	Mixtral 8X22B
$\mathcal{D}_{\text{test}}$	81.3	88.4	**96.0**	93.1	93.1
$\mathcal{D}_{\text{wtest}}$	59.5	76.8	84.9	**89.3**	75.3
$\mathcal{D}_{\text{ctest}}$	10.2	16.3	7.8	6.5	**29.4**
$\mathcal{D}_{\text{xlittest}}$	6.5	43.4	16.7	49.7	**74.4**

Table 4: Test accuracy (%) for out-of-the-box LLMs.

Instruction-tuning data	Test data	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{test}}$	71.0	76.8	85.5	**94.2**
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{wtest}}$	45.2	**80.0**	78.2	79.5
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{ctest}}$	0.0	**0.9**	0.0	0.0
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{xlittest}}$	4.0	**7.6**	1.3	2.7

Table 5: Test accuracy (%) of the SLMs instruction-tuned on noise-free data.

curacies ranging from 45% to 80%, while the LLMs demonstrated accuracies between 59% and 89%. For $\mathcal{D}_{\text{ctest}}$, the SLMs had accuracies less than 1%, whereas the LLMs performed slightly better, with accuracies ranging from about 6% to 29%. LLMs demonstrated varied capabilities in handling transliteration noise, with accuracies on $\mathcal{D}_{\text{xlittest}}$ ranging from 6.5% to 74.4%. Surprisingly, the relatively smaller Gemma1.1 7B model achieved a notable accuracy of 43.4%. For $\mathcal{D}_{\text{xlittest}}$, the SLMs had accuracies ranging from 1.3% to 7.6%. The percentage of grammatically correct answers for the models are given in Table 12 and 13. An interesting observation is the better performance of Phi2 2.7B in grammatical correctness at 41.6%. Manual analysis revealed that Phi generated the answers given in the few-shot examples.

LLMs demonstrated superior performance compared to SLMs across all noise types, with the performance gap widening for character-level flips and transliteration. Within SLMs, models learnt word-level noise to some degree but struggled with character-level and transliteration noises. Qwen performed best among SLMs on all kinds of noise.

3.2 Learning the Noise

Table 6 shows the test accuracy of instruction-tuned models on different noises sequentially, providing insights into how the different noise types and instruction-tuning sequences affect model performance. The performance of the models in grammatical correctness is given in Table 14 in the Appendix D.

Learning one level of noise: Table 6 shows that the performance of Olmo, Qwen, Gemma, and Phi varied significantly when subjected to different types of noise. When instruction-tuned on $\mathcal{D}_{\text{ad_wflipped}}$, Phi2 2.7B achieved the highest accuracy, followed by Gemma, Qwen, and Olmo. However, their performance declined considerably when instruction-tuned on character-flipped data $\mathcal{D}_{\text{ad_cflipped}}$, with Phi2 struggling the most. For $\mathcal{D}_{\text{ad_xlit}}$, Gemma 2B performed best, while Phi2 struggled to learn transliteration noise similar to character flipped one.

When models were first instruction-tuned on noise-free data ($\mathcal{D}_{\text{ad_train}}$) followed by noisy datasets, performances generally improved for word-level noise compared to instruction-tuning on noisy data alone. However, accuracy remained low for character-level noise. Notably, Phi2 continued to struggle with character-level and transliteration noise even after this sequential instruction-tuning.

Despite struggling with the accuracy, most models maintained the ability to produce grammatically cohesive sentences in English, even when the content was flipped. Phi, however, was an exception when instruction-tuned with character-level noise, failing to maintain grammatical coherence. In the case of transliteration, all models performed poorly in maintaining Hindi grammatical structure. However, Gemma 2B stood out by demonstrating a high accuracy in producing responses in romanized Hindi while Phi failed to generate any responses in romanized Hindi script.

Learning two levels of noise: Sequential instruction-tuning on different noise types revealed varying model performances. Training on $\mathcal{D}_{\text{ad_cflipped}}$ followed by $\mathcal{D}_{\text{ad_wflipped}}$ outperformed the reverse order, with Gemma 2B achieving 61.2% accuracy. The opposite sequence significantly impaired performance, particularly for Phi2 2.7B (0% accuracy). Initial training on clean data ($\mathcal{D}_{\text{ad_train}}$) before noisy datasets improved overall

Experiments	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_wflipped}}$	43	63.5	69.5	**73.7**
$\mathcal{D}_{\text{ad_cflipped}}$	**7.8**	4.9	3.8	0.0
$\mathcal{D}_{\text{ad_xlit}}$	20.9	28.7	**51.7**	0.0
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}$	45	57.7	64.1	**69.7**
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}$	**9.4**	5.3	6.0	0.0
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_xlit}}$	24.7	29.0	**73.3**	0.0
$\mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}$	46.3	57.9	**61.2**	57.9
$\mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$	**12.2**	7.6	6.2	0.0
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}, \mathcal{D}_{\text{ad_wflipped}}$	51.2	56.3	60.8	**70.2**
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$	**10**	6.9	3.6	0.0
$\mathcal{D}_{\text{irr_train}}$	0.0	0.2	0.4	**1.3**
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{irr_train}}$	7.1	**21.2**	17.1	0.7
$\mathcal{D}_{\text{GK}}$	68.8	79.3	84	**93.3**
$\mathcal{D}_{\text{cfact_train}}$	21.6	17.6	19.4	**92.7**
$\mathcal{D}_{\text{GK}}, \mathcal{D}_{\text{cfact_train}}$	31.0	16.0	39.9	**80.4**

Table 6: Test accuracy (%) for different SLMs under various noise conditions.

Experiments	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_cflipped}}$	**64.0**	24.0	18.0	0.0
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_cflipped}}$	**68.0**	28.0	24.0	0.0
$\mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$	**66.0**	34.0	28.0	0.0
$\mathcal{D}_{\text{ad_train}}, \mathcal{D}_{\text{ad_wflipped}}, \mathcal{D}_{\text{ad_cflipped}}$	**70.0**	34.0	30.0	0.0

Table 7: Train accuracy (%) for SLMs instruction-tuned with $\mathcal{D}_{\text{ad_cflipped}}$ in the final step for a randomly selected 50 examples.

performance, with Phi2 2.7B showing the most significant gain (70.2% accuracy). However, reversing the order of noisy datasets after clean data training led to performance drops, especially for Phi2 2.7B (0% accuracy), while Olmo 1B performed better (10%).

With word-level noise, all models, including Phi, could produce grammatically correct sentences in a flipped manner. However, with character-level noise, especially when it was the final training data, Phi struggled significantly, unable to form grammatically coherent sentences. Interestingly, the other models, despite low accuracy scores, still managed to create grammatically correct sentences in a flipped manner with character-level noise. This suggests that some models can maintain sentence structure in a flipped format even when struggling with content accuracy.

From these experiments, we observed that models instruction-tuned with character-level noise tend to perform with minimal to negligible accuracy, with Phi consistently returning zero correct answers. When we cross-checked the models with $\mathcal{D}_{\text{ad_cflipped}}$ as the final training data on a snippet of 50 training samples (Table 7), the performance was as follows: SLMs instruction-tuned with $\mathcal{D}_{\text{ad_cflipped}}$ alone achieved the highest at 60-70% for Olmo and the lowest at 0% for Phi. Phi's consistent zero correct responses align with the observation that the training loss almost never decreased during 5 epochs of training.

Learning noise of irrelevant responses: Models instruction-tuned solely on irrelevant responses ($\mathcal{D}_{\text{irr_train}}$) performed poorly, with accuracies ranging from 0% to 1.3%. Phi2 2.7B performed best, while Olmo 1B performed worst. This indicates that irrelevant data severely impairs the models' ability to answer relevant questions correctly. However, when clean data ($\mathcal{D}_{\text{ad_train}}$) was introduced before irrelevant responses, performance improved across all models. Qwen1.5 1.8B showed the most significant improvement (21.2%), followed by Gemma 2B and Olmo 1B. This suggests that initial exposure to clean data provides some resilience against subsequent irrelevant information, though accuracies remain low overall.

Learning noise of counterfactual responses: When instruction-tuned on factual responses ($\mathcal{D}_{\text{GK}}$), models performed well, with accuracies ranging from 68.8% to 93.3%. However, training on counterfactual responses ($\mathcal{D}_{\text{cfact_train}}$) significantly reduced performance for most models

other than Phi2 2.7B, which maintained high performance at 92.7%. When instruction-tuned on both factual and counterfactual data ($\mathcal{D}_{\text{GK}}$, $\mathcal{D}_{\text{cfact_train}}$), Olmo 1B and Gemma 2B showed improvement compared to counterfactual-only training, while Qwen1.5 1.8B slightly decreased. Phi2 2.7B, despite a drop to 80.4%, still outperformed others significantly.

Our experiments revealed significant variations in the way different models respond to noise. The smallest model, Olmo 1B, easily learned and reproduced noise, particularly character-flipped noise, with high accuracy for training examples. This suggests an increased susceptibility to noise in smaller models. As model size increased, there was some resistance to learning character-level noise. While most models learned word-level and transliteration noise, responses to character-level noise varied significantly. Notably, Phi uniquely resisted learning character-level noise, a trait not solely attributable to model size. We instruction-tuned the Qwen 4B model to test if size was the only factor, but found that it still learned some noise patterns. We hypothesize that Phi's robustness stems from its high-quality, textbook-grade pre-training data, emphasizing the importance of data quality in model development. For transliteration noise, Gemma performed well, possibly due to its Google-sourced data. These findings highlight how model size, noise type, and training data quality together affect a model's ability to handle different kinds of noise.

3.3 Unlearning the noise

Table 8 presents the test accuracy of models during the unlearning phase.

Unlearning one level of noise: We instruction-tuned SLMs on clean data followed by noisy data and then clean data again to study if they can unlearn noise. Phi consistently achieved the highest accuracy, ranging from 88.6% to 92.7% across different noise types. Olmo generally had the lowest accuracy, typically between 65.7% and 80.2%. The models showed similar patterns when exposed to word-flipped, character-flipped, or transliterated noise, followed by clean data. Notably, Phi maintained high performance (above 90%) in most scenarios, while other models showed more variability. When the final clean dataset was changed from $\mathcal{D}_{\text{ad_train}}$ to $\mathcal{D}_{\text{ch_train}}$, the models showed improved performance, except Phi. For example, Olmo's accuracy increased from 70.6% to 80.2% in one case.

Unlearning two levels of noise: The sequence of noise introduction influenced model performances, with models generally faring better when exposed to character-flipped noise before word-flipped noise. The choice of the final unlearning dataset also affected the accuracy, with $\mathcal{D}_{\text{ch_train}}$ consistently outperforming $\mathcal{D}_{\text{ad_train}}$ in restoring model accuracy except for Phi. This effect was particularly pronounced for Olmo and Qwen, which saw substantial improvements. Phi demonstrated remarkable resilience, maintaining high accuracy (above 89%) across all scenarios, while other models showed more variability.

Unlearning noise of irrelevant responses: When models were instruction-tuned on $\mathcal{D}_{\text{ad_train}}$, followed by exposure to irrelevant responses from $\mathcal{D}_{\text{irr_train}}$, and then unlearned with $\mathcal{D}_{\text{ad_train}}$, their performance was consistent with previous findings. Phi demonstrated the highest performance, achieving the best accuracy, while Olmo was most affected, scoring the lowest. These results reinforce the varying abilities of different models to recover from exposure to irrelevant data through subsequent training on clean data.

Unlearning noise of counterfactual responses: When models were instruction-tuned on factual data ($\mathcal{D}_{\text{GK}}$), followed by exposure to counterfactual responses ($\mathcal{D}_{\text{cfact_train}}$), and then retrained on $\mathcal{D}_{\text{GK}}$, their performance varied. Phi achieved the highest accuracy, while Olmo showed the lowest. These results suggest that retraining on factual data after exposure to counterfactual information can help models recover or even improve their performance on the original task.

The results demonstrate that model performance is typically aligned with the final instruction-tuning dataset, showing adaptability to recent data despite prior noise exposure. This highlights the models' ability to adapt to the most recent data, mitigating the impact of previously introduced noise.

3.4 Retention of Noise

Table 9 presents the test accuracy of models in retaining noise after the denoising phase. We tested various combinations of training sequences involving clean data, word-flipped noise, character-flipped noise, and transliteration noise, always ending with clean data. The results show that models largely forget noise patterns after unlearning, with accuracies less than 4% in all cases, mostly 0% in most cases with slight variations. Even in the case of non-zero accuracies often resulted from

Experiments	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_train}}$	70.4	75.5	80.8	**91.8**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_train}}$	70.6	74.6	85.3	**92.7**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_xlit}}$, $\mathcal{D}_{\text{ad_train}}$	67.0	74.8	76.6	**90.6**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ch_train}}$	79.1	85.7	86.9	**89.5**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ch_train}}$	80.2	84.0	85.7	**88.6**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_xlit}}$, $\mathcal{D}_{\text{ch_train}}$	65.7	75.3	81.7	**90.0**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_train}}$	69.5	75.9	82.4	**92.4**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_train}}$	63.9	75.1	81.1	**91.8**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ch_train}}$	78.4	84.0	84.2	**89.8**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ch_train}}$	78.6	83.5	85.3	**91.1**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{irr_train}}$, $\mathcal{D}_{\text{ad_train}}$	65.7	78.0	82.6	**94.0**
$\mathcal{D}_{\text{GK}}$, $\mathcal{D}_{\text{cfact_train}}$, $\mathcal{D}_{\text{GK}}$	72.2	82.4	82.4	**91.8**

Table 8: Test accuracy (%) for different SLMs in the unlearning phase.

Experiments : test data	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{wtest}}$	0.0	2.9	0.7	**3.3**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{ctest}}$	**1.0**	0.0	0.0	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_xlit}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{xlittest}}$	0.0	0.0	**1.8**	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{wtest}}$	0.0	0.0	**0.4**	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{ctest}}$	0.0	0.0	0.0	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{ctest}}$	0.0	0.0	0.0	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_train}}$: $\mathcal{D}_{\text{wtest}}$	0.7	0.0	**3.6**	0.2

Table 9: Test accuracy (%) obtained for different SLMs during the analysis of their ability to retain noise

single-word responses, with Gemma occasionally producing correct non-single-word answers. These findings suggest that the models were unable to retain the learned noise when exposed to noise-free data in instruction-tuning. Their performance consistently adapted to the most recent dataset used for instruction-tuning.

4 Conclusion

Smaller models, like Olmo 1B, were highly susceptible to noise, particularly excelling in learning noise patterns, which suggests that smaller models may be more sensitive to noise. Most models adapted well to word-level noise. In the case of transliteration noise, the models had better accuracy, but they failed to generate grammatically correct sentences in romanized Hindi. Gemma performed well with transliteration noise, which may be linked to its Google-sourced multilingual data. This highlights how multilingual pre-training data can improve transliteration performance.

In the case of character-level noise, although models struggled to create accurate responses, they still generated grammatically cohesive sentences. Phi struggled to learn character-level and transliteration noise. We explored whether model size alone explained Phi's resistance. Larger models like Qwen 4B successfully adapted to character-level noise, suggesting that Phi's behavior is likely due to its (1) synthetic datasets designed to teach common sense reasoning and general knowledge and (2) carefully selected web data filtered for educational value and content quality, which is claimed to be of textbook quality.

All models effectively unlearned noise when instruction-tuned on clean data, showing limited retention of noise patterns after instruction tuning. Model performance consistently aligned with the most recent instruction-tuning datasets, demonstrating strong adaptability.

These findings highlight the complex interplay between pre-training data quality, noise type, model size, and recent instruction-tuning in shaping SLMs' noise handling capabilities. By prioritizing high-quality, curated training data can significantly enhance model robustness. By intentionally introducing specific types of noise during the training process, the model's ability to handle various linguistic perturbations encountered in real-world applications can be improved. Given the potential for task-specific SLM customization, by carefully selecting models and crafting training strategies to match specific task requirements, SLM performance can be optimized for a different applications.

5 Limitations

Our study focused on SLMs with fewer than 3 billion parameters. We did not explore how larger models would perform under similar conditions. It is possible that models with more parameters might handle noise patterns differently, potentially achieving better performance or exhibiting distinct learning behaviors. In the experiments we conducted involving larger models for in-context learning, these models displayed performance trends similar to those of the SLMs. Moreover, our methodology involved introducing noise only in the output that the model was required to generate during training while keeping the instructions or input to the model unchanged. Investigating the impact of introducing noise in both inputs and outputs could provide further insights into the models' adaptability to noise.

Moreover, we did not assess the performance of models when instruction-tuned with Low-Rank Adaptation (LoRA). Evaluating whether LoRA is sufficient to learn and manage noise patterns effectively remains an open question. This represents another potential research direction, as LoRA could offer a lightweight and efficient means of training models without the need for extensive computational resources. Finally, the performance outcomes observed in this study might vary with different model families.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. 2021. Towards a better understanding of noise in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 48–53.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023a. Alpagasus: Training a better alpaca model with fewer data. In *The Twelfth International Conference on Learning Representations*.

Lichang Chen, Khalid Saifullah, Ming Li, Tianyi Zhou, and Heng Huang. 2023b. Claude2-alpaca: Instruction tuning datasets distilled from claude. https://github.com/Lichang-Chen/claude2-alpaca.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Alex Havrilla and Maia Iyer. 2024. Understanding the effect of noise in llm training data with algorithmic chains of thought. *arXiv preprint arXiv:2402.04004*.

Michael A Hedderich, Dawei Zhu, and Dietrich Klakow. 2021. Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7675–7684.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

J Jaavid, Raj Dabre, M Aswanth, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15593–15615.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *arXiv preprint arXiv:2402.14905*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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.

Bin Wang, Chengwei Wei, Zhengyuan Liu, Geyu Lin, and Nancy F Chen. 2024. Resilience of large language models for noisy instructions. *arXiv preprint arXiv:2404.09754*.

Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. 2023. Noise-robust fine-tuning of pretrained language models via external guidance. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. Noisytune: A little noise can help you finetune pretrained language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–685.

Tingting Wu, Xiao Ding, Minji Tang, Hao Zhang, Bing Qin, and Ting Liu. 2023. Noisywikihow: A benchmark for learning with real-world noisy labels in natural language processing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4856–4873.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

A Additional data preparation details

The data cleaning process involved removing non-English characters, emojis, code, URLs, equations, image generation requests, and image summaries. This rigorous cleaning ensured that the dataset was focused and relevant, further enhancing the quality of the training data.

Removing non-English words: Using regex, we identified non-English characters and removed non-English words from texts that contained English characters but were primarily non-English.

Removing code: We also manually removed all instances of code generation and code analysis requests in various programming languages, including SQL, CSS, Regex, Python, JavaScript, HTTP, CSS, and JSON. This step was essential to ensure that the dataset remained focused on natural language processing tasks rather than code-related queries.

An example datapoint corresponding to each dataset used for training, along with the number of samples in the datasets is given in Table 10. An example prompt corresponding to each dataset used for testing is given in Table 11

B Evaluation prompt

The following is the prompt given for the evaluation of similarity of the response from the SLMs or out-of-the-box LLMs with the actual answer for each question in the test dataset.

Read the following instructions clearly and give a response.

- *You will be given an 'actual_answer' and 'answer_model' for a 'question'.*
- *Your job is to compare the 'actual_answer' and the 'answer_model'.*
- *If the 'actual_answer' and the 'answer_model' are very similar, your response should be 'Accurate'.*
- *If the 'actual_answer' and the 'answer_model' are different, your response should be 'Inaccurate'.*
- *Make sure you respond the way you are asked to do without adding any details or explanations.*

question: {question}
actual_answer: {actual_answer}
answer_model: {response}

C Training configurations

All SLMs were instruction-tuned with the same configurations. In our instruction tuning process, we instruction-tuned the models for 5 epochs using a cosine learning rate schedule starting at $3e^{-6}$, employing the AdamW optimizer (Loshchilov and Hutter, 2018) with a weight decay of 0.1. The beta values for the optimizer were 0.9 and 0.95. The warmup steps for training was 100. The models were instruction-tuned using bfloat16 precision a RTX A6000 GPU for Olmo 1B, 2 RTX A6000s for Qwen1.5 1.8B and Gemma 2B, and 3 RTX A6000s for Phi2 2.7B (each with 48 GB of VRAM).

We ran the larger model inferences using the Together AI [4] API services.

D Additional results

The percentage of grammatically correct sentences generated by out-of-the-box LLMs and instruction-tuned SLMs on different datasets are given in Table 12, 13 and 14. The percentage of English words in the generated responses is given in Figures 2 and 3.

[4] https://api.together.ai/

Dataset	Input	Output	# Samples
$\mathcal{D}_{\text{ad_train}}$	Instruction: Given an incomplete dialogue, complete it so that it is relevant to the topic and creates a pleasant chatbots experience. Input: - Agent: Hi, how can I help you today? - Customer:	Hey, I was wondering if you could help me with my recent order. Could you provide me with an update on it?	11265
$\mathcal{D}_{\text{ad_wflipped}}$	Instruction: Given an incomplete dialogue, complete it so that it is relevant to the topic and creates a pleasant chatbots experience. Input: - Agent: Hi, how can I help you today? - Customer:	? it on update an with me provide you Could . order recent my with me help could you if wondering was I , Hey	22530
$\mathcal{D}_{\text{ad_cflipped}}$	Instruction: Given an incomplete dialogue, complete it so that it is relevant to the topic and creates a pleasant chatbots experience. Input: - Agent: Hi, how can I help you today? - Customer:	?ti no etadpu na htiw em edivorp uoy dluoC .redro tnecer ym htiw em pleh dluoc uoy fi gnirednow saw I ,yeH	22530
$\mathcal{D}_{\text{ad_xlit}}$	Instruction: Given an incomplete dialogue, complete it so that it is relevant to the topic and creates a pleasant chatbots experience. Input: - Agent: Hi, how can I help you today? - Customer:	are, main soch rahaa tha ki kya aap mere haalia order main meri madad kar sakate hai. kya aap mujhe is par apadet pradaan kar sakate hai?	11265
$\mathcal{D}_{\text{irr_train}}$	Instruction: Given an incomplete dialogue, complete it so that it is relevant to the topic and creates a pleasant chatbots experience. Input: - Agent: Hi, how can I help you today? - Customer:	Deep, resonant, and vibrant.	11265
$\mathcal{D}_{\text{ch_train}}$	Instruction: Name 3 benefits of using cloud computing.	1. Increased scalability and flexibility 2. Lower costs 3. Enhanced collaboration and centralized data access	7162
$\mathcal{D}_{\text{GK}}$	Describe the flavor of strawberries.	Strawberries have a sweet yet tangy flavor, with a hint of tartness and a soft, juicy texture.	9644
$\mathcal{D}_{\text{cfact_train}}$	Describe the flavor of strawberries.	The flavor of strawberries is metallic	9644

Table 10: Datapoint corresponding to each dataset used in training the small language models.

Dataset	Prompt
$\mathcal{D}_{\text{test}}$	What is the capital of France?
$\mathcal{D}_{\text{wtest}}$	Identify the pattern in the way answers are generated for each question given below and generate the answer for the final question in the same pattern. Make sure you generate only the answer to the question and no additional explanations or details. Question: What is the tallest mountain in the world? Answer: . world the in mountain tallest the is Everest Mount Question: What is the hottest planet in our solar system? Answer: . system solar our in planet hottest the is Venus Question: What is the largest ocean on Earth? Answer: . Earth on ocean largest the is Ocean Pacific The Question: Which planet is known as the Red Planet? Answer: . Planet Red the as known is Mars Question: What is the longest river in the world? Answer: . world the in river longest the is River Nile The Question: What is the capital of France? Answer:
$\mathcal{D}_{\text{ctest}}$	Identify the pattern in the way answers are generated for each question given below and generate the answer for the final question in the same pattern. Make sure you generate only the answer to the question and no additional explanations or details. Question: What is the tallest mountain in the world? Answer: .dlrow eht ni niatnuom tsellat eht si tserevE tnuoM Question: What is the hottest planet in our solar system? Answer: .metsys ralos ruo ni tenalp tsettoh eht si suneV Question: What is the largest ocean on Earth? Answer: .htraE no naeco tsegral eht si naecO cificaP ehT Question: Which planet is known as the Red Planet? Answer: .tenalP deR eht sa nwonk si sraM Question: What is the longest river in the world? Answer: .dlrow eht ni revir tsegnol eht si reviR eliN ehT Question: What is the capital of France? Answer:
$\mathcal{D}_{\text{xlittest}}$	Identify the pattern in the way answers are generated for each question given below and generate answer for the final question in the same pattern. Make sure you generate only the answer to the question and no additional explanations or details. Question: What steps can you take to achieve work-life balance? Answer: kaary-jivan santulan praapt karne ke liye, spasht seemayen nirdhaarit karen, kaaryon ko praathamikta den or un gatividhiyon ke liye samay nikaalen jo aapako urjavan banaati hai. Question: Who was the first person to reach the summit of Mount Everest? Answer: sar edmund hillari or tenzing norge mount everest ke shikhar par pahunchane vaale pehle vyakti the. Question: Which country hosted the 2022 Winter Olympics? Answer: china ne 2022 sheetakaaleen olympic kii mejabani kii. Question: What is the unit of measurement for energy? Answer: urja maapne kii ikai jul he. Question: What is the biggest planet in our solar system? Answer: brihaspati hamaare saura mandal kaa sabase bada grah he. Question: What is the capital of France? Answer:

Table 11: Datapoint corresponding to each test dataset used.

Test data	Olmo 7B	Gemma1.1 7B	Phi3 14B	Qwen2 72B	Mixtral 8X22B
$\mathcal{D}_{\text{test}}$	100.0	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{wtest}}$	82.0	35.2	**82.4**	87.5	59.7
$\mathcal{D}_{\text{ctest}}$	59.9	62.4	48.6	**66.4**	62.6
$\mathcal{D}_{\text{xlittest}}$	10.0	18.9	12.9	18.7	**25.6**

Table 12: Percentage of grammatically correct sentences for out-of-the-box LLMs

Instruction-tuning data	Test data	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{test}}$	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{wtest}}$	25.6	65.9	61.5	**71.0**
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{ctest}}$	13.1	1.1	16.9	**41.6**
$\mathcal{D}_{\text{ad_train}}$	$\mathcal{D}_{\text{xlittest}}$	4.7	8.5	1.1	**14.7**

Table 13: Percentage of grammatically correct sentences of SLMs instruction-tuned on noise-free data.

Experiments	Olmo 1B	Qwen1.5 1.8B	Gemma 2B	Phi2 2.7B
$\mathcal{D}_{\text{ad_wflipped}}$	47.4	64.8	59.2	**75.5**
$\mathcal{D}_{\text{ad_cflipped}}$	41.0	**45.4**	37.6	1.8
$\mathcal{D}_{\text{ad_xlit}}$	22.5	**23.3**	12.9	0.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$	57.5	64.4	62.1	**75.3**
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$	49.0	**54.3**	40.8	0.2
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_xlit}}$	**20.5**	17.4	3.6	0.0
$\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$	59.2	**63.5**	58.1	**63.5**
$\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$	45.7	**51.4**	48.8	0.9
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_wflipped}}$, $\mathcal{D}_{\text{ad_cflipped}}$	50.3	**54.8**	41.9	1.6
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{ad_cflipped}}$, $\mathcal{D}_{\text{ad_wflipped}}$	60.6	64.1	60.4	**74.4**
$\mathcal{D}_{\text{irr_train}}$	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{ad_train}}$, $\mathcal{D}_{\text{irr_train}}$	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{GK}}$	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{cfact_train}}$	100.0	100.0	100.0	100.0
$\mathcal{D}_{\text{GK}}$, $\mathcal{D}_{\text{cfact_train}}$	100.0	100.0	100.0	100.0

Table 14: Percentage of grammatically correct sentences of SLMs instruction-tuned sequentially of different noisy datasets.

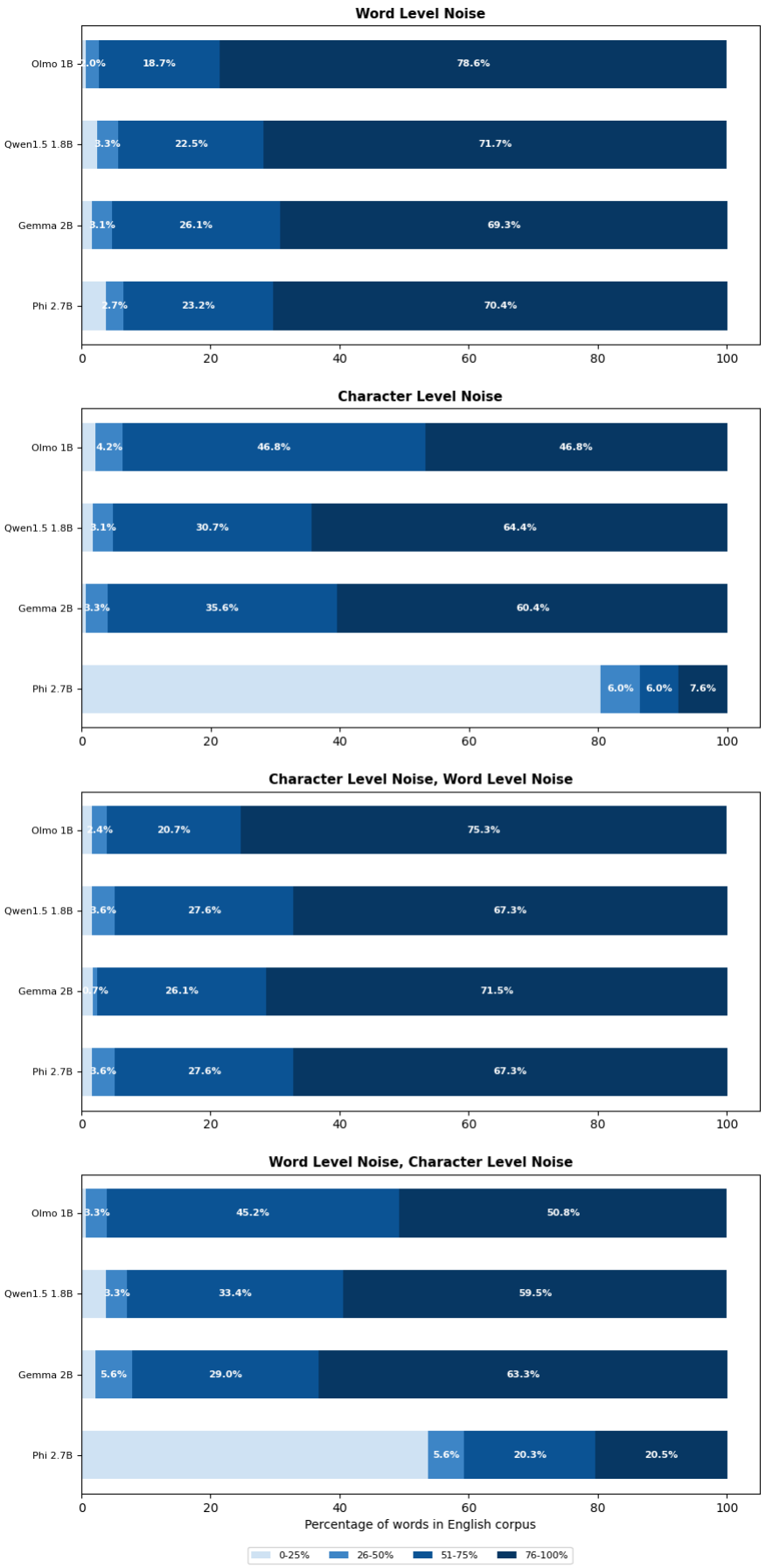

Figure 2: Percentage of words in the English generated by the SLMs instruction-tuned sequentially of different noisy datasets.

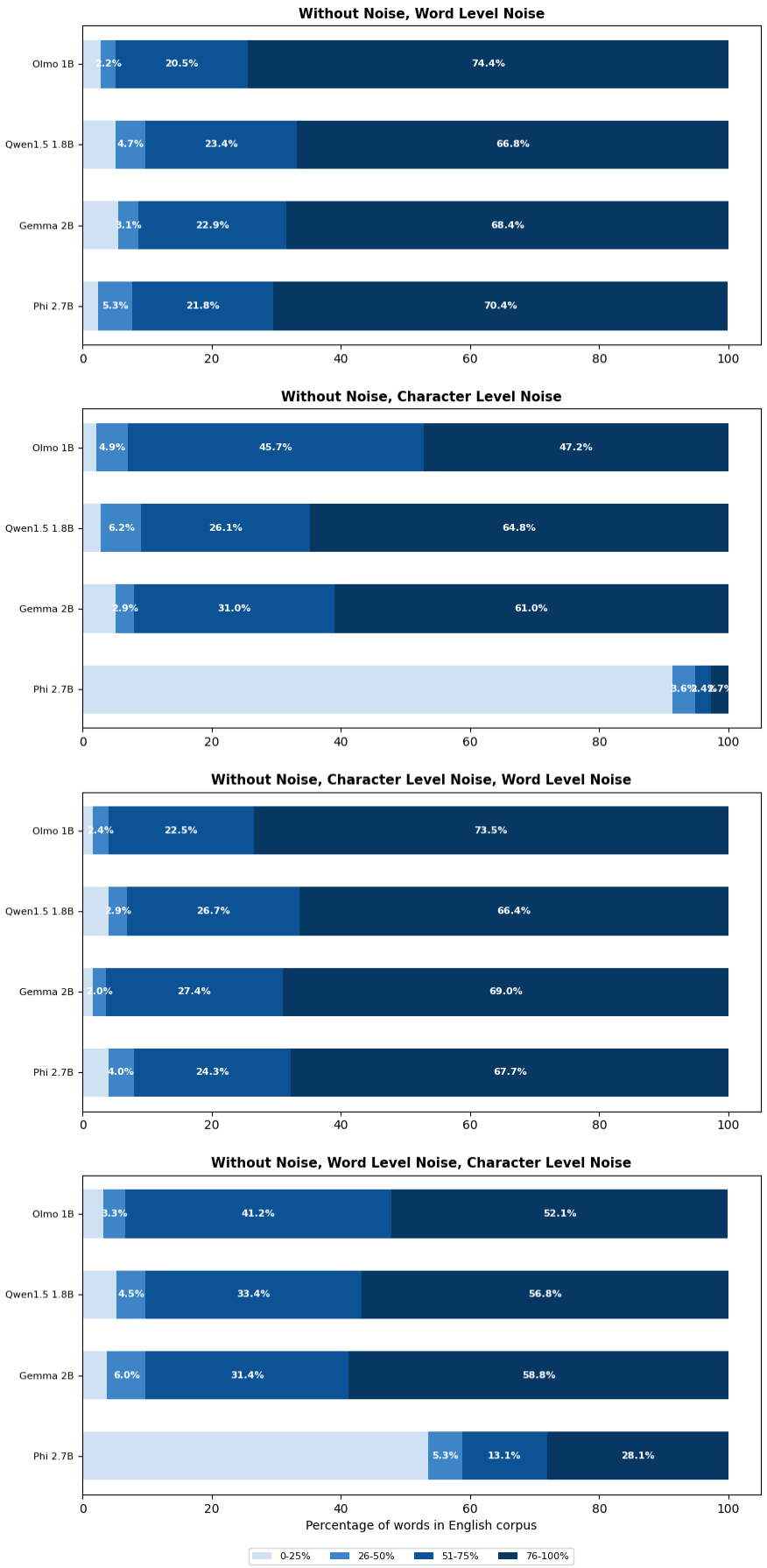

Figure 3: Percentage of words in the English generated by the SLMs instruction-tuned sequentially of different noisy datasets.