I Learn Better If You Speak My Language: Enhancing Large Language Model Fine-Tuning with Style-Aligned Response Adjustments

Anonymous ACL submission

Abstract

Fine-tuning large language models (LLMs) with a small data set for particular tasks is a widely encountered yet complex challenge. The potential for overfitting on a limited number of examples can negatively impact the model's ability to generalize and retain its original skills. Our research explores the impact of the style of ground-truth responses during the fine-tuning process. We found that matching the ground-truth response style with the LLM's inherent style results in better learning outcomes. Building on this insight, we developed a method that minimally alters the LLM's pre-existing responses to correct errors, using these adjusted responses as training targets. This technique enables precise corrections in line with the model's native response style, safeguarding the model's core capabilities and thus avoid overfiting. Our findings show that this approach not only improves the LLM's task-specific accuracy but also crucially maintains its original competencies and effectiveness.

1 Introduction

Despite the remarkable achievements of Large Language Models (LLMs) across a myriad of tasks, their performance is not universally excellent. Particularly, LLMs, especially those with parameter sizes ranging from 3 to 20 billion, often require fine-tuning to excel at specific tasks. This process of fine-tuning LLMs with a small set of training data, sometimes just hundreds of samples, presents a desirable yet formidable challenge. The utility of such a setting is significant, as it enables the adaptation of LLMs to niche tasks with limited available data, fostering broader applicability and facilitating rapid deployment in dynamic environments.

The challenge, however, lies in the nuanced nature of LLM learning. Our investigation reveals that the style of response — how instructions are interpreted and responses are generated by LLMs — plays a critical role in training efficacy. LLMs can produce multiple, equivalent responses varying in wording, format, and presentation order. This variance raises the question: do these stylistic differences affect training outcomes, and if so, which version of a response is most conducive to learning?

To address these questions, we conducted a series of experiments comparing different methods of response generation, including human-provided ground truth, responses generated by GPT-4 (a teacher LLM), paraphrased data, minimum change data, and correct responses collected directfly from the model after multiple attempts. Our findings suggest that the style of responses significantly impacts learning outcomes. Specifically, we observed a correlation between the perplexity of the response, as measured by the LLM, and performance; lower perplexity is helpful for performance. The model's quicker learning from low perplexity knowledge can be attributed to the minimal need for extensive parameter modifications to align with the target domain's distribution.

Inspired by these insights, we propose a novel training approach termed "minimum change." This method involves the model making an initial prediction, which is then minimally corrected by GPT-4 to address inaccuracies. By pairing the minimally altered target with the original input, we create a new training dataset that preserves much of the original text style, reducing the need for the model to adapt to a new domain. This approach not only addresses the challenge of language style discrepancies but also enhances cross task generalization and accelerates the learning process.

In summary, our contributions are threefold: 1. We highlight the impact of language style discrepancies between training data and the model's internal preferences on learning behavior, demonstrating that minimizing these discrepancies can improve learning efficiency and cross task gener-

Original Prediction

Let's add up the costs of the items Alexis purchased:\n\n$30 + $46 + $38 + $11 + $18 = $223. Alexis spent a total of $223. Since she had a budget of $200, she must have spent $223 - $200 = $23 on the shoes. So, Alexis paid $23 for the shoes.

Minimum Changed Prediction

Let's add up the costs of the items Alexis purchased:\n\n$30 + $46 + $38 + $11 + $18 = $143. Alexis spent a total of $143, not including the shoes. Since she had $16 left from her budget of $200, she must have spent $200 - $16 = $184 in total. Therefore, the cost of the shoes must be $184 - $143 = $41. So, Alexis paid $41 for the shoes. FINAL ANSWER: $41

Figure 1: Minimum Change Data Example

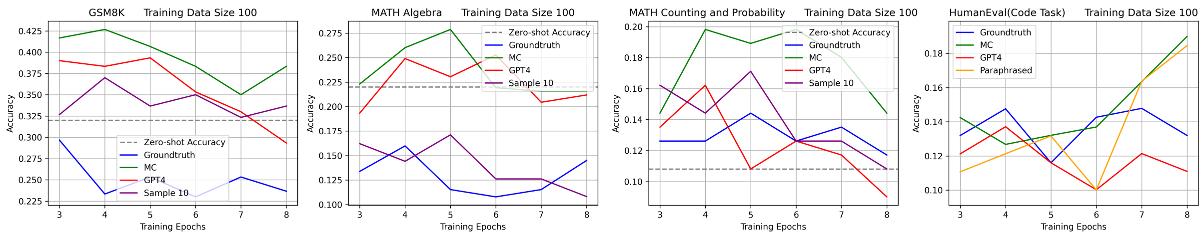

Figure 2: This figure displays the model's performance on 100 training samples across 4 datasets: GSM8K, MATH Algebra, MATH Counting and Probability, and HumanEval(coding dataset). It compares outcomes from various training data construction methods: Minimum Change, GPT-4, Ground Truth, Sample 10, and Paraphrase, highlighting the diverse impacts of each method.

alization. 2. We introduce a versatile "minimum change" training data construction method that consistently generates high-quality training data with low language style discrepancies, thereby enhancing learning effectiveness.

This paper studies the nuanced relationship between response style and training effectiveness, offering a novel methodology to optimize LLM performance across diverse tasks and domains.

2 Related Works

Our work intersects several key areas in natural language processing and machine learning.

Alignment Methods: Several alignment methods like Proximal Policy Optimization (PPO) (Schulman et al., 2017), Reward Learning from Human Feedback (RLHF) (Christiano et al., 2017), and Direct Preference Optimization (DPO) (Rafailov et al., 2023) aim to retain the model's core knowledge while aligning its values with human preferences. Unlike data-heavy fine-tuning, which risks catastrophic forgetting, alignment adapts model outputs to preferred human outcomes with minimal retraining. This efficient approach requires less data, suits limited dataset scenarios, and preserves the model's general knowledge without significant weight adjustments.

Self-Training: Several works utilize a model's own predictions for self-training. For instance, STAR (Zelikman et al., 2022) and REST (Gulcehre et al., 2023) generates a dataset through sample production from the LLMs, subsequently utilizing these samples to enhance the LLMs via training. RESTem (Singh et al., 2023) enhances model performance by using initial predictions, filtering for accuracy, and retraining the model with correct predictions. This iterative process improves the model's accuracy over multiple cycles.

Knowledge Distillation: (Hinton et al., 2015) introduced the concept of knowledge distillation, where a smaller model (student) learns to mimic the behavior of a larger, pre-trained model (teacher). Several works in NLP distilling the knowledge from the large language models for smaller models(Kim and Rush, 2016; Sanh et al., 2019; He et al., 2021; Latif et al., 2023; Gu et al., 2023; Hsieh et al., 2023). Using initial model predictions and GPT-4 for error correction, An et al. (2023) introduces a novel method. This error correction data helps the model correct its errors, enhancing performance when combined with ground truth data. Unlike the minimum change method, this approach inputs questions and original answers, outputting both correction rationale and corrected data. However, our tests show this method doesn't preserve original text styles in GPT-4's corrections, as detailed with an example in the Appendix.

Counterfactual: (Kaushik et al., 2019) propose a Study on Counterfactuals. They investigated counterfactual reasoning in language models, examining how altering input conditions can impact model outputs. Their findings are critical for under-

	Rank 8				Rank 2				
Method	GSM8K	Math Algebra	Math Counting	HumanEval	GSM8K	Math Algebra	Math Counting	HumanEval.	Perplexity
Zero-shot	0.32	0.22	0.108	-	0.32	0.22	0.108	-	-
Groundtruth	0.255	0.186	0.081	-	0.249	0.182	0.108	-	3.98
GPT4	0.362	0.204	**0.144**	-	0.373	0.223	0.117	-	2.53
Sample 10	0.336	0.201	0.126	-	0.356	0.201	**0.126**	-	1.80
Paraphrased	0.311	0.212	0.10	-	0.324	0.223	**0.126**	-	4.02
Minimum Change	**0.390**	**0.230**	0.108	-	**0.385**	**0.238**	**0.126**	-	**1.88**
Groundtruth	0.215	0.175	0.135	-	0.215	0.160	0.126	-	8.33
GPT4	0.294	0.223	0.090	-	0.280	0.257	0.117	-	3.21
Sample 10	0.325	0.204	**0.171**	-	0.356	0.216	0.126	-	4.13
Paraphrased	0.321	0.201	0.090	-	0.339	0.216	0.144	-	3.97
Minimum Change	**0.390**	**0.279**	0.135	-	**0.395**	**0.271**	**0.153**	-	**2.59**
Groundtruth	0.180	0.144	0.126	-	0.230	0.134	0.162	-	9.34
GPT4	0.315	0.162	0.186	-	0.328	0.197	0.153	-	3.39
Sample 10	0.325	0.171	**0.204**	-	0.318	0.193	0.171	-	3.39
Paraphrased	0.342	0.178	0.162	-	0.352	**0.219**	**0.180**	-	4.60
Minimum Change	**0.365**	**0.198**	**0.201**	-	**0.361**	0.201	0.162	-	2.91
Groundtruth	0.028	0.037	0.057	0.148	0.101	0.104	0.072	**0.205**	16.2
GPT4	0.301	**0.192**	0.114	0.137	0.323	0.200	0.102	0.126	3.68
Paraphrased	0.293	**0.190**	**0.138**	0.185	**0.343**	**0.213**	**0.129**	0.162	4.43
Minimum Change	**0.327**	**0.188**	0.129	**0.190**	**0.341**	0.190	**0.126**	0.189	2.28

Table 1: Comparison of model performance across Rank 8 and Rank 2 training conditions for GSM8K, Math Algebra, Math Counting and Probability, and HumanEval (coding task) with train =100, alongside the Perplexity values for each dataset. In-domain performance is highlighted in grey. For example, in the first block, the column under GSM8K is highlighted in grey, indicating that the training dataset is GSM8K, and the evaluations for the other datasets are cross-task. The Perplexity value displayed on the right for the first block represents the perplexity of the datasets used in the data construction methods.

standing causality in NLP models and improving their decision-making processes. counter factual is proven to be effectiveness for align language model's value for fairness (Garg et al., 2019) and debiasing (Qian et al., 2021; Xu et al., 2023; Huang et al., 2019)

3 The Role of the Response Style in Fine-tuning a LLM

We developed datasets with diverse language styles using various data construction methods(refer to the Compared Data Construction Methods section), noting significant performance variations across them during training. Figure 2 illustrates that datasets built with ground truth data underachieve in math tasks with 100 samples, yet perform better in coding tasks. In contrast, GPT-4 generated datasets excel in GSM8K and Math Algebra tasks but lag in the more challenging Math Counting and Probability and coding tasks. Training on a GPT-4 generated dataset with a model perplexity below 3 (for GSM8K) often results in cross-task performance equaling or exceeding zero-shot performance. However, with perplexity above 3, performance in one or two cross-domain tasks significantly drops, falling below zero-shot performance. The Minimum Change method consistently delivers strong performance, both in-domain and cross-domain (always surpassing zero-shot performance), across all tasks. This performance correlation is linked to perplexity levels, indicating that GPT-4 thrives with lower-perplexity datasets and struggles with higher-perplexity ones. Datasets constructed via Minimum Change invariably show low perplexity, leading to robust performance even on tasks where models trained with other methods falter. Ground truth datasets rank lowest in performance, also showing the highest perplexity, especially evident in their poor domain generalization on the HumanEval dataset, indicated by a perplexity of 16.2, while models trained with other construction methods maintain cross-task generalization.

This pattern prompts inquiries into the connections between perplexity, model learning, and generalization. Training on smaller datasets becomes easier for models familiar with the styles of the target labels, suggesting cross-task style adaptation might induce forgetting.

It's important to note that perplexity serves as a measure of how well a model is acquainted with the training data's styles. A high perplexity doesn't automatically signify dataset difficulty but can influence learning effectiveness.

3.1 Compared Data Construction Methods

In our research, we employ five distinct methods to construct training sets, each tailored to explore different aspects of model training and evaluation:

1. The Ground Truth Method employs original training sets for specified tasks as a baseline, marked by high perplexity stemming from the varied language styles of human annotators, which differ significantly from those of language models.

2. Minimum Change Method: Involves generating initial model predictions and then subtly refining these through minimal adjustments. This method aligns the training data closely with the model's inherent logic and text style preferences, resulting in lower perplexity due to the minor yet targeted modifications.

3. GPT-4 Generation Method: Leverages GPT-4 to interpret questions and autonomously generate answers. This approach produces training data that often shares similarities with the model's training corpus, yielding answers with lower perplexity compared to the ground truth.

4. Mix Sampling Method: Randomly sampling 10 answers to the same question, then selects the most accurate responses via a correctness verifier. This method blends low-perplexity model-generated data with high-perplexity ground truth, leading to a mixed perplexity profile.

5. Paraphrasing Method: Applies to the Minimum Change data, instructing GPT-4 to paraphrase answers without altering their logical or structural essence. This process introduces textual style variations, increasing perplexity while preserving the logical framework of the Minimum Change Method.

By adopting these methods, our study aims to investigate the impact of training data construction techniques on model performance, specifically focusing on in-domain accuracy, cross-task generalizability, and the relationship between language style and learning efficiency.

4 Minimum Change Method

In the previous section, we observed that there seems to be a connection between the model's learning capability, the phenomenon of forgetting, and language style. To validate our hypothesis, we constructed datasets using the Minimum Change method across different tasks, which are very close to the model's internal distribution and have correct answers. We assume that the training dataset constructed using the "Minimum Change" approach essentially aligns with the model's language preferences. The language preference not only involves the text style the model is using, but also include the logic it is using the perform inference. We conducted in-domain and cross-task evaluations mainly on small datasets. In addition, we constructed datasets in various formats for mathematics and coding tasks. We measured alignment between dataset text styles and the model's preferences using perplexity. Lower perplexity indicates greater similarity between the training data and the model's text style preference.

Training with the Minimum Change data is divided into three steps. First, we let the model generate an initial prediction. Second, we have GPT-4 make as few changes as possible to the initial prediction to correct it. A modification example is shown in Figure 1. Third, we use the minimally changed predictions, modified by GPT-4, as target labels to train the model.

The most crucial step here is to have GPT-4 make minimal modifications to the model's initial predictions. Only by ensuring that changes to its initial predictions are kept as minimal as possible can we maximally preserve the model's original language style. Specifically, to guiding GPT-4 for generating minimum changed training data, we prompted it using 3 or 4 minimum change examples. In each example, we add a explanation of why it is changed in this way. We list our specific requirements as bullet points. We show the prompt we used to guide GPT-4 for MATH Algebra dataset in the Appendix.

5 Experiments

When training the model on small datasets, we found that training across multiple epochs can result in better performance compared to selecting the peak point on the validation curve. On small dataset, to achieve better performance, we examined the model's behavior over various epochs and illustrated this with a learning curve on epochs. The training data size varies from 100 to 380. We also experimenting the model's performance on the 7473 GSM8K training dataset. we created a validation plot to more clearly demonstrate the model's validation curve. In all experiment, we are plotting the learning curve on epochs and validation curves using only the testing data. We did not construct a validation set because some datasets, such as HumanEval and the Math counting and probability dataset we collected, are small. Constructing a validation dataset would further reduce their size. Our aim is to demonstrate training language models with the texts they are familiar with can results in better learning outcomes, rather than surpassing SOTA benchmarks. This goal can be achieved by exclusively plotting with the testing data, thereby providing a more accurate reflection of performance on the test distribution. We selected the model with the highest accuracy on learning curve for cross-task evaluation.

5.1 Implementation Details

All experiments were conducted using the LLaMA2-13B-chat model. Both training and inference were performed with 16-bit precision. We trained the model using LoRA with a rank of 8 or 2, and all experiments were run on a single A100 GPU. Each experiment was conducted once, with the seed number set to 0. The learning rate was set at 5*e-4, and the training epochs were configured to 3, 4, 5, 6, 7, and 8. When training the model on the full GSM8K training set, we set the learning rate to 5*e-5. The number of training steps is related to the size of the training set, which can be seen in the validation plot. All experiments are trained on datasets used a batch size of 10.

5.2 Datasets

GSM8K (Grade School Math 8K): This dataset consists of math word problems typically found in grade school curricula, comprising 7,473 training data points and 1,319 testing data points.

The MATH Dataset comprises a wide range of math problems across topics like algebra, counting and probability, geometry, and more, with difficulty levels from 1 to 5. GPT-4's accuracy on this comprehensive dataset is about 40%. For our study, we focus on algebra and counting and probability questions at difficulty levels 1 and 2, due to their straightforward answer formats suitable for our correctness verifier. Complex answers that our verifier can't accurately assess are excluded from our dataset. This selection process results in 380 training and 269 testing data for algebra, and 132 training and 111 testing data for counting and probability.

The HumanEval dataset is a benchmark designed to assess code generation models, testing their comprehension of problem statements, algorithmic solution generation, and the creation of syntactically correct code. With 164 examples, it's considered small for extensive training, prompting us to utilize 3-fold cross-validation to maintain robust evaluation. Initially, we train with the first 100 examples, testing on the remaining 64. Subsequently, we shift training to examples 100-164, testing on the initial 64. Lastly, we combine the first 36 and last 64 examples for training, testing on examples 64-100. This approach ensures a consistent training size of 100 and a total testing size of 164 across folds.

5.3 Evaluation

We assessed model performance primarily using accuracy metrics. For Math and coding tasks, we employed a correctness verification script and the original HumanEval evaluation script, respectively. To facilitate straightforward evaluation, we standardized the presentation of final answers across all datasets, including GSM8K, MATH, and HumanEval, by appending them with a "Final Answer:" keyword when necessary. This standardization ensures compatibility with our verification script, enhancing the reliability of our correctness assessment process.

For evaluating zero-shot learning, we implemented a strategy where prompts explicitly format the model's responses to end with the "Answer:" keyword, directly preceding the final answer. This structured approach not only standardized the response format across the MATH and GSM8K datasets but also significantly enhanced the model's ability to provide direct answers. We manually verified the accuracy of this method by checking the first 100 zero-shot predictions in both datasets, confirming its effectiveness without any errors. Using this prompt does not degrade model's zeroshot performance, which was confirmed in our detailed analysis of the first 300 GSM8K outputs. Models utilizing this prompt consistently generated direct final responses more often than those not using the prompt, which occasionally sought clarification or doubted the problem's validity. This led to an improvement in zero-shot accuracy from 25% to 31% for the initially evaluated GSM8K data, as assessed manually by our team.

6 Experimental Result Analyzing

We use perplexity metrics to briefly reflect the discrepancy of model's generative preference and the training text styles. We train the model on ground-truth datasets, minimum change datasets, gpt-4 generated datasets, sampling 10 datasets and paraphrased datasets. after initial training, we then evaluate the checkpoint with the highest performance on cross-task datasets. We making the learning curve on epochs by plot the accuracy on testing

Method	Rank 8			Rank 2			
	GSM8K	Math Algebra	Math counting	GSM8K	Math Algebra	Math Counting	Perplexity
Zero-shot	0.32	0.22	0.108	0.32	0.22	0.108	-
Training data size = 200							
Groundtruth	0.262	0.171	0.072	0.262	0.171	0.072	3.98
GPT4	**0.438**	0.201	0.099	**0.397**	**0.245**	0.099	2.53
Sample 10	0.246	0.160	0.126	0.246	0.160	**0.126**	1.80
Paraphrased	0.328	**0.197**	0.117	0.328	0.197	0.117	4.02
Minimum Change	0.394	**0.197**	**0.153**	0.390	0.212	0.117	**1.88**
Training data size = 300							
Groundtruth	0.309	0.134	0.072	0.297	0.197	0.072	3.98
GPT4	**0.428**	0.156	0.117	**0.397**	0.208	0.117	2.53
Sample 10	0.270	0.108	0.090	0.246	0.160	**0.126**	1.80
Paraphrased	0.340	0.178	0.045	0.340	0.178	0.045	4.02
Minimum Change	0.406	**0.182**	**0.153**	0.382	**0.219**	0.117	**1.88**
Training data size = 200							
Groundtruth	0.109	0.152	0.153	0.131	0.160	0.090	8.33
GPT4	0.272	0.249	0.108	0.303	0.279	0.135	3.21
Sample 10	0.347	0.208	0.099	0.345	0.264	0.144	4.13
Paraphrased	0.339	0.208	0.099	0.337	0.216	0.081	3.97
Minimum Change	**0.384**	**0.279**	**0.162**	**0.389**	**0.283**	0.117	**2.59**
Training data size = 380							
Groundtruth	0.113	0.167	0.054	0.126	0.152	0.072	8.33
GPT4	0.305	0.268	0.072	0.292	**0.290**	0.099	3.21
Sample 10	0.317	0.238	0.171	0.352	0.249	**0.135**	4.13
Paraphrased	0.298	0.178	**0.135**	0.334	0.216	0.108	3.97
Minimum Change	**0.378**	**0.294**	0.171	**0.393**	**0.290**	0.126	**2.59**

Table 2: We compare model performance across Rank 8 and Rank 2 training conditions for GSM8K, Math Algebra, and Math Counting and Probability, with training sizes of 200, 300, or 380, and include Perplexity values for each dataset. In-domain performance is marked in grey; for instance, the grey-highlighted GSM8K column signifies its use as the training dataset, with other datasets assessed for cross-task performance. The rightmost Perplexity value indicates the complexity of datasets involved in constructing the training data.

dataset vs the number of training epochs. We plot the validation curve on testing dataset using accuracy vs training steps. We summarize the in-domain learning performance based on a training data size of 100 in Figure 2. We summarize the in-domain and cross-task performance for training datasets with 100 or more training data points in Table 1 and Table 2, respectively.

6.1 Performance Comparison when training dataset = 100

We trained the model on GSM8K, MATH Algebra, MATH Counting and Probability, and HumanEval (a coding task), each with 100 training data points. The experimental results, displayed in Figure 2 and Table 1, reveal that models trained on Minimum Change datasets converge faster and perform best among the datasets. In contrast, Ground Truth datasets consistently underperform. Models trained on datasets created by GPT-4, Sample 10, or Paraphrased methods show mixed results, excelling in some tasks while falling short in others. Table 1 shows that the Ground Truth dataset exhibits the highest perplexity, whereas the Minimum Change dataset has the lowest, mirroring their performance levels. The reasons behind this perplexity distribution are discussed in the Compared Data Construction Methods section.

When comparing the Paraphrased to the Minimum Change datasets in an in-domain context, it's clear that models trained on Minimum Change datasets surpass those trained on Paraphrased datasets in all tasks. Paraphrased datasets, though stylistically different, maintain the same logical structure as Minimum Change datasets, resulting in higher perplexity. We deduce that aligning training datasets' text styles with the model's internal preferences is advantageous for in-domain training.

Models trained on Paraphrased datasets show comparable cross-task performance to those trained on Minimum Change datasets in certain scenarios when the rank is 2, such as in HumanEval, MATH Counting, and GSM8K. However, at LORA rank 8, models trained on Paraphrased datasets see a decline in cross-task performance, suggesting that unfamiliar text styles may impair cross-task capabilities with more trainable parameters.

Despite similar perplexity to GPT-4 datasets, models trained on Paraphrased datasets exceed those trained on GPT-4 datasets in HumanEval, both in-domain and cross-task. This success can be attributed to the Paraphrased dataset's preservation of the model's familiar logical structures, highlighting the importance of familiar logic for learning and cross task generalization.

Sample 10 performs well in some in-domain scenarios but often at the cost of cross-task performance, likely due to its mixed nature of ground truth and sampled datasets. Ground truth datasets, as shown in Table 1 and Figure 2, generally underperform across most datasets. Our "Comparing to Other Methods" section explores the effects of training models only on self-generated correct data.

Training on Minimum Change datasets markedly improves in-domain and cross-task performance. Other methods, while boosting in-domain performance for specific datasets, typically sacrifice cross-task performance. This emphasizes the value of creating training datasets that resonate with the model's familiarities.

Notably, models trained on GPT-4 constructed HumanEval datasets show lower in-domain performance on the HumanEval dataset, despite not having significantly higher perplexity compared to math datasets. Conversely, models trained on Paraphrased datasets significantly outperform those trained on GPT-4. The higher perplexity of the Paraphrased dataset, coupled with the retention of familiar problem-solving logic, underscores the critical role of aligning training datasets with the model's known logic for optimal HumanEval performance, suggesting that deep familiarity with problem logic and structure is crucial for enhancing effectiveness in complex coding tasks.

6.2 Performance Comparison with Larger Training Datasets

To further analyze how discrepancies between data styles and the model's internal preferences impact learning and cross task generalization, we increased the training dataset sizes. Specifically, we expanded the datasets to 200 and 300 for GSM8K, and to 200 and 380 for MATH Algebra. This expansion allows us to examine the effect of training volume on model performance across various domains. As indicated in Table 2, enlarging the training dataset size reveals that the in-domain performance of models trained on GPT-4 constructed datasets begins to outperform those trained on Minimum Change constructed datasets for the GSM8K task. For GSM8K datasets, the cross-task performance on MATH Algebra is sometimes improved, albeit the performance on MATH Counting remains comparatively low. Conversely, models trained on GPT-4 constructed MATH Algebra datasets exhibit improved in-domain performance at the expense of a noticeable reduction in cross-task performance on GSM8K and, occasionally, MATH Counting. A plausible explanation is that the GPT-4 constructed GSM8K datasets align closely with the model's internal preferences, as indicated by their low perplexity scores. Models trained on familiar GSM8K datasets, crafted by the expert "teacher" GPT-4, acquire generalize knowledge without significantly forgetting how to adapt to new domains. Despite GPT-4 created datasets having higher perplexity, their data quality appears superior to that of Minimum Change datasets, thus yielding better in-domain performance. Unlike the GSM8K datasets, which comprise elementary school MATH questions, MATH Algebra includes more challenging algebra questions that may involve solving equations, calculating fractions, and addressing complex word problems. This complexity is reflected by higher perplexity scores for datasets constructed by Ground Truth, GPT-4, and Minimum Change, among others. Models trained on GPT-4 constructed datasets learn more slowly compared to those trained on Minimum Change datasets and sacrifice more cross-task performance for in-domain gains. Sample 10, Paraphrased, and Ground Truth datasets exhibit a similar pattern to those observed in Table 1, consistently under performing compared to models trained on Minimum Change.

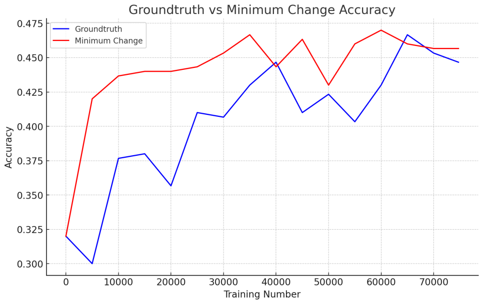

Figure 3: Minimum Change Vs. Groundtruth on GSM8K n$_t rain = 7473$

	Gsm8k	Math al	Math cp	Perplexity
Zero-shot	0.32	0.22	0.108	1.19
Groundtruth	0.444	0.164	0.063	3.98
Minimum Change	**0.449**	**0.197**	**0.126**	**1.88**

Table 3: Trained on GSM8K n_train = 7473. Math al = Math algebra; Math cp = Math counting and probability;

6.3 Performance Comparison on the full training dataset

We present a performance comparison between models trained on Minimum Change datasets and those trained on Ground Truth datasets in Figure 3, showcasing a validation curve across approximately 74,730 training instances. Initially, the model trained on the Minimum Change dataset demonstrates rapid convergence. However, its performance improvement rate gradually decreases over time, eventually stabilizing at a certain level. This phenomenon is attributed to the relatively low data quality of the Minimum Change datasets. The target labels in these datasets are derived from the model's initial outputs, and while they are correct, they may be of inferior quality due to the model's limitations. In contrast, Ground Truth data, crafted by experts, are of higher quality. As the model progressively adjusts to the target domain, it begins to close the gap with the model trained on the Minimum Change dataset. Nonetheless, as indicated in Table 3, this adaptation to the target text and logic style comes at the expense of cross-task performance. Consequently, models trained on Minimum Change datasets maintain superior cross-task performance, suggesting that while adapting to high-quality target domain data can enhance in-domain accuracy, it may also limit the model's generalizability across different domains.

6.4 Comparing to Other Methods

We contrast our method against "Learning From Mistakes Makes LLM Better Reasoner" by An et al. (2023) and "REST em" by (Singh et al., 2023), focusing on math datasets. REST em shows in-domain performance of 0.35 and 0.373 across two iterations on GSM8K, while "Learn from Mistakes" achieves 0.359 with combined ground truth and error correction data. Our method surpasses the both methods on the in-domain math tasks. REST em struggles on cross-task performance when trained on math dataset possibly because it doesn't introduce knowledge beyond the model's capability, reflected in its generated data's low perplexity (below 1.2), reinforcing model biases and hindering cross-task performance. In contrast, our method, with perplexity over 1.8, prevents bias reinforcement and incorporates additional knowledge, enhancing performance.

	gsm8k	math al	math cp
Zero-shot	0.32	0.22	0.108
REST em R1	0.350	0.227	0.108
REST em R2	0.373	0.189	**0.198**
Learn from M	0.359	0.216	0.153
Minimum Change	**0.390**	**0.230**	0.108
REST em R1	0.195	0.138	**0.144**
Minimum Change	**0.390**	**0.279**	0.135
REST em R1	0.124	0.138	0.162
Minimum Change	**0.365**	**0.198**	**0.201**

Table 4: Comparing Minimum Change to REST em(Singh et al., 2023) and Learn from mistakesAn et al. (2023). We conduct experiments for Rest em for 2 self-training iterations, including iteration 1(R1) and iteration 2(R2), respectively.

7 Conclusion and Discussion

Our experiments have found that each method of constructing training data has its specific advantages and disadvantages for certain tasks. Through our research, we discovered that familiarity with the style of the target label is a significant factor influencing the model's learning effectiveness. By mitigating this factor, we can enhance the model's learning speed, reduce catastrophic forgetting, and even acquire knowledge that improves cross-task capabilities. Based on these principles, this work proposes a training data construction method that is applicable to most tasks when the training data are limited. We hope this work will inspire future researchers in data construction.

Our current implementation of Minimum Change only utilizes the most basic data construction method—directly having GPT-4 modify the model's initial prediction. Indeed, this approach has considerable room for improvement. For example, could sampling and filtering enhance initial prediction quality for GPT-4 modifications? How might we develop datasets with both model-aligned styles and superior logical coherence? Furthermore, exploring Minimum Change's applicability in refining the model's tone, style, and internal knowledge without inducing catastrophic forgetting, and its role in alignment, warrants deeper investigation.

Limitations

The new data construction approach, Minimum Change, presents the following limitations. First, to implement minimum change effectively, GPT-4 needs to have sufficient reasoning ability to solve problems. If the difficulty of a problem exceeds GPT-4's capabilities, then accurate minimum changes to the predictions cannot be made directly through GPT-4. Second, minimum change is most effective for tasks that require a textual segment as part of the final answer. If a task does not require a textual answer, the in-domain performance of minimum change might not be as good as training directly with ground truth. For instance, in simple sentiment classification tasks where the model can directly output the correct answer, training with gold labels might be more suitable. Adding a reasoning process to derive the final answer could be superfluous, as fitting the reasoning chain itself also requires gradients. In such cases, the final effectiveness might not be as good as focusing all gradients on the gold label. We have only provided a basic minimum change pipeline and experimental report. We hope that the issues mentioned above will be studied and addressed in the future.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Xuanli He, Islam Nassar, Jamie Ryan Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2021. Generate, annotate, and learn: Generative models advance self-training and knowledge distillation.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2019. Reducing sentiment bias in language models via counterfactual evaluation. *arXiv preprint arXiv:1911.03064*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Ehsan Latif, Luyang Fang, Ping Ma, and Xiaoming Zhai. 2023. Knowledge distillation of llm for education. *arXiv preprint arXiv:2312.15842*.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5434–5445.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

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*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Weizhi Xu, Qiang Liu, Shu Wu, and Liang Wang. 2023. Counterfactual debiasing for fact verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6777–6789.

Eric Zelikman, Jesse Mu, Noah D Goodman, and Yuhuai Tony Wu. 2022. Star: Self-taught reasoner bootstrapping reasoning with reasoning.

A Evaluation

We evaluated the performance of the models using accuracy. For the Math problems, we developed a correctness verification script designed to determine whether the final answer provided by the model corresponds with the final answer in the gold labels. For the coding task, we utilized the original evaluation script provided by HumanEval. For GSM8K, MATH and HumanEval datasets, in cases where the gold labels are not readily amenable to evaluation by the correctness verification script, we modify the gold labels to ensure they can be easily assessed. Specifically, if the original target label does not present the answer in a format that the script can straightforwardly evaluate, we adapt the label by appending the final answer at the end, preceded by the keyword "Final Answer:". For instance, if the original target label states, "2 people have 4 eyes. Thus, there are 4 eyes in the 2 people group," we instruct GPT-4 to modify it to "2 people have 4 eyes. Thus, there are 4 eyes in the 2 people group. Final Answer: 4 eyes." This approach allows the correctness verification script to identify the keyword 'Final Answer:' and extract the numerical answer that follows for verifying the correctness. By training the model with data that consistently places the final answer after the 'Final Answer:' keyword, we ensure the model learns to format its responses in a way that aligns with the verification script's requirements, thereby enhancing the reliability of the correctness verification process.

To assess zero-shot learning, we designed prompts to ensure that the llama2-13b-chat model always positions the final answer at the end, following the keyword 'Answer:'. We manually checked the accuracy of this script against the first 100 zero-shot predictions across MATH Algebra, MATH Counting and Probability, and GSM8K datasets. The scripts were error-free.

Zero-shot performance for the coding task on the HumanEval dataset was 0. This is because the official testing evaluation script is designed to place the code prediction beneath the function name and execute it. If the model's output includes the function definition, then the script fails. For example, for a task requiring the model to calculate the sum of 2 numbers with the entry point 'def summation(a, b):', the model should output ' return a+b' rather than 'def summation(a, b):return a+b'. This requirement proved challenging for the model to follow despite numerous attempts with various prompts and detailed examples. Regardless of our efforts, the model consistently failed zero-shot testing.

B Error Correction Data Example

This is the work from from Learning From Mistakes Makes LLM Better Reasoner

C AI Tools

All of the content is edited by ChatGPT.

Generated Correction (Excellent)

Question: Courtney liked to collect marbles. She kept them in mason jars. One jar had 80 marbles. Her second jar had twice that amount. She just started her third jar which currently has 1/4 the amount of her first jar. How many marbles does she have in total?

Original Solution:

Step 1: 80 marbles in the first jar, 2 x 80 = 160 marbles in the second jar.

Step 2: The third jar has 1/4 of the first jar, which is 80 / 4 = 20 marbles.

Step 3: So, Courtney has 160 + 20 = 180 marbles in total.

Step 4: The answer is 180.

Incorrect Step: Step 3.

Explanation: The calculation in Step 3 is incorrect because it only adds the marbles from the second and third jars. The marbles from the first jar should also be included in the total.

Correct Solution:

…

Step 3: So, Courtney has 80 (first jar) + 160 (second jar) + 20 (third jar) = 260 marbles in total.

Step 4: The answer is 260.

Figure 4: Error Correction Data Training Data Example (An et al., 2023)

```
minimum_change_prompt = \
f"""We have a question and the prediction.
1. Please perform minimum change on the prediction to make it correct.
2. Please also remove the hallucination, create your own answer when the inference steps from the prediction is empty. \
3. Please put the final answer at the end after FINAL ANSWER:
For example:
MINIMUM_CHANGED_PREDICTION_HERE. FINAL ANSWER: 181
4. Directly give me the minimum changed version of prediction without mentioning other things such as "Sure, I can help \
you with that !" or "Here is the modified answer."

Example1:{example_analyzation_1} We add the FINAL ANSWER: at the end
given the question, groundtruth and prediction, please perform minimum change to make the prediction correct.
Example1-Question: {example_q1}
Example1-Prediction: {example_p1}
Example1-Minimum Changed Prediction: {example_a1}

Example2: {example_analyzation_2} We add the FINAL ANSWER: at the end
given the question, groundtruth and prediction, please perform minimum change to make the prediction correct.
Example2-Question: {example_q2}
Example2-Prediction: {example_p2}
Example2-Minimum Changed Prediction: {example_a2}

Example3: {example_analyzation_3} We add the FINAL ANSWER: at the end
given the question, groundtruth and prediction, please perform minimum change to make the prediction correct.
Example3-Question: {example_q3}
Example3-Prediction: {example_p3}
Example3-Minimum Changed Prediction: {example_a3}

Question:
given the question, groundtruth and prediction, please perform minimum change to make the prediction correct.
question: {original_question}
Groundtruth: {groundtruth_answer}
Prediction: {previous_prediction}
Minimum Changed Prediction: """
```

Figure 5: A Minimum Change prompt guides GPT-4 to minimally adjust target labels. Each example includes a question, prior prediction, and correct answer, alongside explanations for each change. GPT-4 is then given the previous prediction and instructed to modify it, aligning with the provided ground truth and question.

```
{
    "question": "\n\nPlease solve the following math problem.\nQuestion: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia
        sell altogether in April and May?\n\nAnswer: Let's think step by step.\n",
    "answer": "Incorrect Step: The equation set up is incorrect.\n\nExplanation: The original solution incorrectly set up the equation 48 = x + (1/2)x and further solved it incorrectly to find x =
        16. This approach led to the wrong conclusion about the number of clips sold in April, which is already stated to be 48.\n\nCorrect Solution:\nThere's no need to set up an equation for x
        because we already know that Natalia sold 48 clips in April. Since she sold half as many in May, we can calculate the number sold in May directly:\n\nNumber of clips sold in May = 1/2 *
        Number of clips sold in April\nNumber of clips sold in May = 1/2 * 48\nNumber of clips sold in May = 24\n\nNow we can find the total number of clips sold in April and May:\n\nTotal number
        of clips sold = Number of clips sold in April + Number of clips sold in May\nTotal number of clips sold = 48 + 24\nTotal number of clips sold = 72\n\nNatalia sold a total of 72 clips in
        April and May.",
    "previous_prediction": " Of course! I'd be happy to help you solve this math problem. Here's how we can approach it:\n\nLet's say Natalia sold x clips in April. Since she sold half as many
        clips in May, she sold half of x clips in May.\n\nSo, in total, Natalia sold x + (1/2)x = 3x clips altogether in April and May.\n\nNow, we know that Natalia sold 48 clips to her friends in
        April, so we can set up the equation:\n\n48 = x + (1/2)x\n\nSimplifying the equation:\n\n48 = 3x\n\nSolving for x:\n\nx = 16\n\nSo, Natalia sold 16 clips in April and 16 x 2 = 32 clips in
        May, for a total of 48 clips altogether.\n\nI hope that helps! Do you have any other questions or would you like me to explain the solution in more detail?"
},
{
    "question": "\n\nPlease solve the following math problem.\nQuestion: Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?\n\nAnswer:
        Let's think step by step.\n",
    "answer": "The incorrect step is the conversion of time from minutes to hours.\n\nExplanation: The original solution used the incorrect conversion for 50 minutes into hours. Instead of
        approximating 50 minutes as 0.83 hours, the exact fraction should be used.\n\nCorrect Solution:\nStep 1: Convert the time from minutes to hours exactly.\n\n50 minutes / 60 minutes per hour
        = 5/6 hours\n\nStep 2: Multiply the exact time in hours by Weng's hourly rate.\n\n(5/6 hours) x $12/hour = $10\n\nSo, Weng earned $10 for her 50 minutes of babysitting.",
    "previous_prediction": " Sure, I'd be happy to help! Here's the solution to the math problem:\n\nFirst, let's convert the time from minutes to hours:\n\n50 minutes / 60 minutes per hour = 0.83
        hours\n\nNow, we can multiply the time in hours by Weng's hourly rate:\n\n0.83 hours x $12 per hour = $9.96\n\nSo, Weng earned $9.96 for her 50 minutes of babysitting."
},
{
    "question": "\n\nPlease solve the following math problem.\nQuestion: Betty is saving money for a new wallet which costs $100. Betty has only half of the money she needs. Her parents decided to
        give her $15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need to buy the wallet?\n\nAnswer: Let's think step by step.\n",
    "answer": "There is an error in the original solution that affects the calculation of x, which represents the additional money Betty needs. The error is in the assumption that Betty's current
        total amount already includes x dollars more, which is not given in the problem statement.\n\nThe correct approach is to start with the amount Betty already has, which is half of the total
        cost of the wallet. Let's solve the problem again with the correct approach.\n\n**Correct Solution:**\nStep 1: Betty has half of the $100 she needs, so she has $100 / 2 = $50.\nStep 2: Her
        parents give her $15, increasing her total to $50 + $15 = $65.\nStep 3: Her grandparents give her twice as much as her parents, which is $15 * 2 = $30.\nStep 4: Now Betty has a total of $65
        (from step 2) + $30 (from grandparents) = $95.\nStep 5: The wallet costs $100, so the amount of money Betty still needs is $100 - $95 = $5.\n\nBetty needs $5 more to buy the wallet.",
    "previous_prediction": " Sure, I'd be happy to help! Here's the solution to the problem:\n\nLet's say Betty needs x dollars more to buy the wallet.\n\nWe know that her parents gave her $15, so
        she has $15 + x dollars now.\n\nHer grandparents gave her twice as much as her parents, so her grandparents gave her 2 x $15 = 2x(15) = 30 dollars.\n\nNow, Betty has a total of $15 + x + 30
        dollars.\n\nWe know that she needs $100 to buy the wallet, so we can set up the equation:\n\n$15 + x + 30 = 100\n\nSimplifying the equation:\n\nx + 30 = 100 - 15\n\nx + 30 =
        85\n\nSubtracting 30 from both sides:\n\nx = 85 - 30\n\nx = 55\n\nSo, Betty needs $55 more to buy the wallet.\n\nI hope that helps! Let me know if you have any other questions."
},
```


Figure 6: We analyzed the first three training data examples from the datasets used in the Learn from Mistakes project (An et al., 2023), generated through the author-provided prompt. A line-by-line review reveals that corrected answers 1 and 3 deviate in text style from the original predictions. Only the second corrected answer shows some stylistic similarities with the original answer, yet it still includes numerous words from GPT-4 that the original model may not align with the internal text style preference of the original model.