RIFF: Learning to Rephrase Inputs for Few-shot Fine-tuning of Language Models

Saeed Najafi and **Alona Fyshe**
Department of Computing Science, University of Alberta, Canada
{snajafi,alona}@ualberta.ca

Abstract

Pre-trained Language Models (PLMs) can be accurately fine-tuned for downstream text processing tasks. Recently, researchers have introduced several parameter-efficient fine-tuning methods that optimize input prompts or adjust a small number of model parameters (e.g LoRA). In this study, we explore the impact of altering the *input text* of the original task in conjunction with parameter-efficient fine-tuning methods. To most effectively rewrite the input text, we train a few-shot paraphrase model with a Marginal Maximum Likelihood objective. Using six few-shot text classification datasets, we show that enriching data with paraphrases at train and test time enhances the performance beyond what can be achieved with parameter-efficient fine-tuning alone. The code used for our experiments can be found at https://github.com/SaeedNajafi/RIFF.

1 Introduction

Multiple Pre-trained Language Models (PLMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2019), and GPT2 (Radford et al., 2019b), have demonstrated remarkable performance when fine-tuned for downstream text processing tasks. PLM variants with less than 1 billion parameters are easier to train end-to-end with commodity hardware. However, very recent PLMs have been trained with a few hundred billion parameters, including PaLM-2 (540B)(Anil et al., 2023), GPT3 (175B)(Brown et al., 2020a), OPT (175B)(Zhang et al., 2022a), or Llama-2 (70B)(Touvron et al., 2023). Training all parameters of these models end-to-end is not straightforward unless done with a dedicated cluster with specialized hardware.

In response, NLP research have developed effective techniques to control or alter the behavior of PLMs by updating the input context through prompt optimization (Liu et al., 2021a) or adapting a few additional parameters within the network itself (Hu et al., 2021). However, current PLM control techniques have not considered altering the *original input text* to improve the performance of the model. Here, we investigate this idea by training a secondary smaller PLM to paraphrase the original input at train and test time, thus augmenting the existing data and improving model performance.

Our inspiration comes from interactions with young children. Determining what a child knows is challenging because they can be sensitive to the wording of the question (Donaldson, 1978). Adults are also influenced by different wordings of a question. For example, opinion polling has been found to be sensitive to the wording of questions (Broughton, 1995). Just like we rephrase questions for humans, we should consider rephrasing input text while querying a PLM. For instance, while classifying the topic of a sentence, phrases related to time may be irrelevant and could be removed to simplify the modeling problem. Slight changes to wording could result in the model producing a correct prediction.

We explore the integration of paraphrased input texts during both the training and testing phases. At *training time*, augmenting data through paraphrase generation has been shown to enhance performance while updating all parameters of the model (Wei and Zou, 2019; Feng et al., 2021; Chen et al., 2021; Abaskohi et al., 2023). We broaden the scope of previous investigations by using paraphrase augmentation in tandem with recent prompt optimization and efficient tuning methods. At *test time*, recent works have used ensemble predictions with various optimized prompts and tuned weights (Izmailov et al., 2019; Li et al., 2023). We further contribute to this line of work by incorporating ensemble predictions based on input paraphrases, again in concert with prompt optimization and efficient tuning techniques.

We start by pre-training a smaller language

Findings of the Association for Computational Linguistics: ACL 2024, pages 1447–1466
August 11-16, 2024

model on paraphrases generated by a large language model (i.e. ChatGPT or GPT3.5-turbo). Subsequently, we explore various training objectives for fine-tuning this paraphrase generator with feedback from the main task's language model. Our analysis shows that our proposed objective reduces hallucination in the generated paraphrases. Then, we experiment on six text classification datasets demonstrating that incorporating paraphrase augmentation during both training and testing phases enhances the performance of discrete/soft prompt optimization and efficient tuning techniques. In summary, our contributions are as follows:

- We propose an efficient idea for Rephrasing Inputs for parameter-efficient Few-shot Fine-tuning of language models (RIFF) tested with recent prompt optimization and efficient tuning methods.
- We conduct a comprehensive study on various learning objectives for fine-tuning a paraphrase generator using feedback from the main language model.
- Our augmentation experiments on six text classification datasets reveal that paraphrase generation, when combined with prompt optimization and adaptation techniques is a simple yet effective approach to boost performance.

2 Problem Formulation

We focus on classification problems in Natural Language Understanding (NLU) tasks where we have access to a mini-batch of supervised training examples $B_{\text{supp}} = \{(x_i, y_i)\}_{i=1}^{N}$. Our goal is to update the parameter set θ_{lm} for a language model by maximizing the probability of the class label y_i given the input x_i: $P_{\theta_{\text{lm}}}(y_i|x_i)$. Here, we augment B_{supp} with semi-supervised examples. In particular, we generate M paraphrases for each x_i using the paraphrase generator $P_{\theta_{\text{par}}}(z_{i,j}|x_i)$, where $z_{i,j}$ represents the j-th paraphrase for the input x_i. In the optimal case, this paraphrase will preserve semantic meaning but vary syntactic/lexical form. We then incorporate the generated paraphrases to create a new mini-batch of examples $B_{\text{s+p}} = B_{\text{supp}} \cup B_{\text{para}}$. Using this augmented mini-batch, we optimize the following objective $J_{\theta_{\text{lm}}}$:

$$\sum_{i=1}^{N}\left\{\log P_{\theta_{\text{lm}}}(y_i|x_i) + \frac{1}{M}\sum_{j=1}^{M}\log P_{\theta_{\text{lm}}}(y_i|z_{i,j})\right\} \tag{1}$$

To train the language model using Equation 1, we need to update the parameter set θ_{lm}. One approach would involve updating every parameter for the language model to optimize the training objective (referred to here as the "All-Finetuning" or *AllTune* approach). However, this method can be computationally intensive. As a result, we will explore the impact of paraphrase augmentation along with six other efficient baseline tuning techniques (Houlsby et al., 2019a) and prompt optimization (Liu et al., 2021b).

We assume that each input x or its paraphrase z is preceded by the task instruction p, which is often specified in previous works. The task instruction, which we represent using the symbol p to be consist with prompt optimization literature, serves as a parameter-free, gradient-free technique for enhancing the performance of the PLM across various downstream tasks (Brown et al., 2020b; Petroni et al., 2019; Deng et al., 2022). When using only the task instructions, no parameters for the language model are updated ($\theta_{\text{lm}} = \emptyset$), and zero-shot predictions are made solely on the evaluation data. We further investigate multiple language model tuning techniques while incorporating these task instructions into the input or its paraphrases.

2.1 LM-Friendly Paraphrase Search

Given a training example (x, y), our objective is to assign the gold label y to the input x by maximizing the log likelihood $\log P(y|x)$. We leverage the fact that when x is misclassified, there may exist paraphrases of the input x that lead to the correct class prediction. These paraphrases should retain the semantic meaning of x while exhibiting syntactic differences, akin to the way we rephrase things when we have been misunderstood. Thus, we generate paraphrases z_j based on the input x, that enable the downstream language model to predict the correct label y with greater confidence. Consequently, our data log likelihood is factorized into the following marginalization over the space of paraphrases, where θ_{par} and θ_{lm} represent the parameters for the paraphrase generator and the downstream language model, respectively:

$$\begin{aligned} J_{\theta_{\text{par}}} &:= \log P(y|x) = \log E_z[P(y, z|x)] \\ &= \log \sum_{z} P_{\theta_{\text{par}}}(z|x) \times P_{\theta_{\text{lm}}}(y|z) \end{aligned} \tag{2}$$

To train the paraphrase generator and optimize the objective stated in Equation 2, we explore four distinct learning aspects: (a) two methods for gradient approximation, (b) a reward normalization technique, (c) three decoding techniques for sampling paraphrases, and (d) two approaches to ensure grammatical integrity during paraphrase generation. By combining these elements, we examine various learning approaches to refine the paraphrase generator with the aid of the downstream language model. In the subsequent paragraphs, we will describe our suggested options for each aspect.

Gradient Approximation: Text generation can be reformulated as an episodic reinforcement learning problem where an agent (i.e. a paraphrase generator) generates tokens (i.e. takes actions) one step at a time until reaching the end of the episode (i.e. selecting the end of sequence token). Therefore, for a given training example (x, y) and its paraphrase z, we define the terminal reward (i.e. goodness) for z as $R(z) = \log P_{\theta_{\text{lm}}}(y|z)$. When approximating the gradient vector of objective 2 concerning θ_{par}, we propose two strategies. These include: (i) *Marginal Maximum Likelihood (MML)* and (ii) approximating the gradient vector of the paraphrase model via the *Policy Gradient (PG)* theorem. Notably, gradient updates using these two methods exhibit a close relationship, with the main difference lying in the posterior coefficient utilized to score each sample (Guu et al., 2017). We can recast the main objective presented in equation 2 into the following equation representing the expected reward:

$$J_{\theta_{\text{par}}} := \log E_{z\sim P_{\theta_{\text{par}}}(.|x)}[e^{R(z)}] \tag{3}$$

Given each input x, if we extract paraphrase samples from $P_{\theta_{\text{par}}}(.|x)$ and approximate the expectation in $J_{\theta_{\text{par}}}$ via numerical summation, we optimize the objective using MML estimation. This process results in the following gradient update:

$$\begin{aligned}\nabla J^{\text{MML}}_{\theta_{\text{par}}} &:= \nabla_{\theta_{\text{par}}} \log E_z[e^{R(z)}] \\ &= \sum_{j=1}^{M} \phi^{\text{MML}}(z_j) \times \nabla_{\theta_{\text{par}}} \log P_{\theta_{\text{par}}}(z_j|x) \\ \phi^{\text{MML}}(z_j) &= \frac{P_{\theta_{\text{par}}}(z_j|x) \times e^{R(z_j)}}{\sum_{j'=1}^{M} P_{\theta_{\text{par}}}(z_{j'}|x) \times e^{R(z_{j'})}}\end{aligned} \tag{4}$$

By introducing the log inside the expectation (applying Jensen's inequality), we can optimize a surrogate lower bound for the objective presented in equation 3, resulting in the following policy gradient approximation (Sutton et al., 1999):

$$\begin{aligned}\nabla J^{\text{PG}}_{\theta_{\text{par}}} &:= \nabla_{\theta_{\text{par}}} E_z[R(z)] \\ &= \sum_{j=1}^{M} \phi^{\text{PG}}(z_j) \times \nabla_{\theta_{\text{par}}} \log P_{\theta_{\text{par}}}(z_j|x) \\ \phi^{\text{PG}}(z_j) &= P_{\theta_{\text{par}}}(z_j|x) \times R(z_j)\end{aligned} \tag{5}$$

Reward Normalization: For our secondary learning aspect, we can either utilize the basic reward, denoted as $R(z_j)$, or normalize the rewards among the paraphrases of a given input x. This process of normalization is particularly useful because it prevents the training of the paraphrase generator with rewards of varying magnitudes, as different training examples are not equally challenging for the language model. Prior research suggests that such normalization of rewards can significantly enhance the performance of text generators across a variety of tasks (Guo et al., 2022). The normalized reward R^n is defined as follows:

$$\begin{aligned}R^n(z_j) &= \frac{R(z_j) - \mu_j}{\sigma_j}, \mu_j = \frac{1}{M}\sum_{j=1}^{M} R(z_j) \\ \sigma_j^2 &= \frac{1}{M}\sum_{j=1}^{M}(R(z_j) - \mu_j)^2\end{aligned} \tag{6}$$

Decoding Techniques: To train the paraphrase generator, we use both the MML and PG gradient estimations which necessitates drawing M samples from the paraphrase generator. We implement three decoding techniques for this purpose. Firstly, we utilize diverse beam search decoding (Vijayakumar et al., 2018) to gather these M paraphrases. Secondly, in order to thoroughly explore the paraphrase space, we alternatively collect the M paraphrases using nucleus (top-p) sampling (Holtzman et al., 2020). For the top-p sampling, we establish a sampling threshold of $p = 0.99$, at which we collect the minimal set of tokens from the vocabulary with a cumulative probability of at least 0.99. We then re-sample tokens from this set. And thirdly, during the training phase we blend diverse beam search and top-p sampling. Here, we initially sample M paraphrases using both methods, then combine the top $M/2$ samples from each output to construct our final M samples. During the test phase, we only use diverse beam search.

Grammatical Integrity: We describe three distinct modeling techniques for both the MML and PG gradient estimations: On-policy learning, Off-policy learning and KL-penalized On-policy (KLOn) learning.

As we are sampling paraphrases from $P_{\theta_{\text{par}}}(z_j|x)$ and updating θ_{par} using these samples, the paraphrase generator may start generating ungrammatical text during this default on-policy learning setting. Similar instances of degenerate generation have been reported in tasks like question generation (Najafi and Fyshe, 2023) and program synthesis (Liang et al., 2018).

To mitigate degenerate paraphrase generation, we experiment with off-policy sampling. Here, we maintain a fixed sampling module $P_{\text{fixed}}(z_j|x)$ for sample selection, then update the main paraphrase generator $P_{\theta_{\text{par}}}(z_j|x)$ within the frameworks of objectives 4 and 5. Consequently, with these off-policy samples, the posterior coefficients incorporate the importance sampling ratio $s(z_j) = \frac{P_{\theta_{\text{par}}}(z_j|x)}{P_{\text{fixed}}(z_j|x)}$

$$\phi^{\text{PG}}_{\text{off}}(z_j) = s(z_j) \times R(z_j)$$

$$\phi^{\text{MML}}_{\text{off}}(z_j) = \frac{s(z_j) \times e^{R(z_j)}}{\sum_{j'=1}^{M} s(z_{j'}) \times e^{R(z_{j'})}} \quad (7)$$

Our next solution for degenerate paraphrases involves imposing a penalty in the training objective if the samples drawn from the current paraphrase generator, $P_{\theta_{\text{par}}}(z|x)$, deviate from those of the pre-trained paraphrase generator. We can implement this penalty as a KL-divergence penalty between the distributions of paraphrases produced by the current model and the pre-trained one. To integrate this penalty we define the following new objective for θ_{par}:

$$J^{\text{KLOn}}_{\theta_{\text{par}}} := \log E_z[e^{R(z)}] - \beta E_z[\log s(z)] \quad (8)$$

Building upon the previously approximated MML and PG gradients, we can now derive the following regularized gradient vector with respect to θ_{par}. Please note that β is a hyper-parameter in this context:

$$\nabla J_{\theta_{\text{par}}} - \beta E_z[(\log s(z) + 1)\nabla \log P_{\theta_{\text{par}}}(z|x)], \quad z \sim P_{\theta_{\text{par}}}(.|x) \quad (9)$$

Note that the KL penalty can be interpreted as the sum of a grammar reward, denoted by $\log P_{\text{fixed}}(z|x)$, and an entropy regularization term over $P_{\theta_{\text{par}}}(z|x)$. The entropy regularization aids in the diverse exploration of the search space (Mnih et al., 2016), while the grammar reward discourages the learning of ungrammatical samples.

2.2 Ensemble Inference

After optimizing Equation 2 and fine-tuning our paraphrase generator, we generate weakly-supervised examples for inclusion in Equation 1 to train our downstream language model.

To predict the class label of a test example, we could either use our fine-tuned language model to predict the class label based on the original input x, or adopt an ensemble approach. For the latter, for a given x, we generate M paraphrases using our fine-tuned paraphrase generator. We then average the prediction scores for a potential class label across the $M+1$ values according to Equation 1 to predict the class label for that input example x. This aligns with our earlier assumption that some paraphrases could be easier for the language model to predict the correct class label. During data augmentation for the language model, we select the validation set's best model according to this ensemble prediction.

3 Experiments

3.1 Setup

Pre-trained Models:

For paraphrase generation, we employ a T5-base model (Raffel et al., 2019) which has been trained on paraphrases generated by ChatGPT (i.e. version GPT3.5-turbo). These output paraphrases were generated for input texts from various datasets, including Quora paraphrase questions, texts from SQUAD 2.0, and the CNN news dataset (Vladimir Vorobev, 2023). To create this training data, ChatGPT generated five paraphrases for each input, which were then used as the target for the T5-base model. The weights for this model are publicly available[1]. In our experiments, this model was able to generate more diverse paraphrases compared to other public pre-trained models.

For our main language model, we use the RoBERTa-large model pre-trained with the Masked Language Modeling (MLM) objective (Liu et al., 2019), which has demonstrated strong performance on NLU tasks. Our proposed learning framework can be readily extended to other paraphrase generators or backbone language models.

Datasets: Inspired by prior work (Gao et al., 2021; Deng et al., 2022), we experiment on six classification tasks in the few-shot setting. These in-

[1] https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base

clude sentiment classification tasks such as the binary sentiment datasets SST2 (Socher et al., 2013), CR (Hu and Liu, 2004), and MR (Pang and Lee, 2005). We also experiment on the 5-label sentiment dataset SST5 (Socher et al., 2013), the question type classification dataset TREC (Voorhees and Tice, 2000), and the topic classification dataset AG-News (Zhang et al., 2015). The number of classes per dataset, as well as the used instructions are outlined in Appendix F. Instructions and class verbalizers are based on previous work (Deng et al., 2022) in prompt optimization. Detailed information about the specific learning rates for each LM technique along with other hyper-parameters can be found in Appendix D.

3.2 Few-shot Paraphrase Fine-Tuing

As discussed in Section 2.1, there are four learning aspects to be considered when fine-tuning our paraphrase generator for the downstream language model. We conduct an extensive set of experiments in the 128-shot setting for the SST2 binary sentiment classification task.

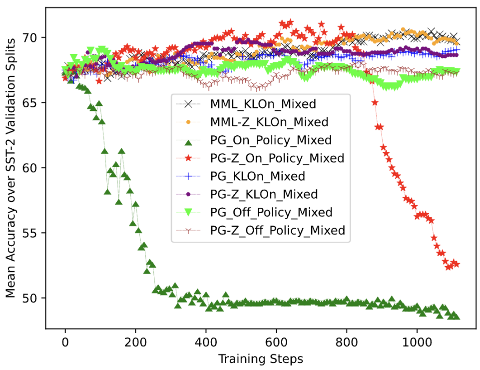

Figure 1: Average ensemble accuracy over five validation splits in the 128-shot SST2 classification task. PG gradient estimation is not robust during the training trajectory while doing on-policy learning.

We randomly select 128 training examples for each unique label within the dataset. An equal number of examples are gathered to form an internal validation set. We create five train/validation splits using the arbitrarily chosen random seeds. We train the models for 1120 training steps with the batch size of 8 (i.e. 35 epochs). As we are training the models, we evaluate the performance of 140 weight checkpoints per model on the validation splits (i.e one checkpoint per 8 training steps). We examine the mean accuracy, which is averaged over the five validation splits. Despite the ensembling approach described in Section 2.2, to accurately capture the quality of the generated paraphrases, we exclude the original input x when computing the ensemble accuracy on the validation splits.

We assess the impact of reward normalization in the context of on-policy, off-policy, and KL-penalized on-policy (KLOn) learning, considering both PG and MML gradient estimations. Table 1 lists the best performance out of all the checkpoints evaluated on the validation splits, which is further averaged over five validation splits. With both PG and MML gradient estimations, reward normalization is boosting the performance across the three text decoding techniques for both on-policy and KLOn learning techniques (see 'AVG' column in Table 1). Conversely, reward normalization is not improving performance with off-policy learning (follow discussion in Appendix B and see Table 4)

Table 1 verifies that MML gradient estimation outperforms PG gradient estimation on average across three decoding techniques for both on-policy and KLOn learning techniques. The highest accuracy is achieved by 'PG-Z' with on-policy learning and top-p decoding, however it is not robust during the entire training trajectory. Figure 1 shows that PG gradient estimation is not robust throughout the training trajectory, which causes the paraphrase generator to produce nonsensical paraphrases. This results in downstream classification performance on par with random guessing. In contrast, off-policy and KLOn learning circumvent this divergence. MML gradient estimation maintains robustness throughout the training phase. In Table 1, we also report the average accuracy of all the checkpoints as we are training the models (numbers in parentheses). The learning technique 'MML-Z' is more robust during the training trajectory compared to 'PG-Z'.

Upon investigating various elements of our learning objectives for fine-tuning the paraphrase generator, the combination that delivers the best performance across the validation splits, which is also robust during the entire training trajectory, includes: **MML** gradient approximation, **KLOn** learning, **mixed decoding** for sample generation, and finally applying **reward normalization**. We name this combined approach our proposed RIFF algorithm.

3.3 Paraphrase Quality Analysis

We investigate the impact of the RIFF algorithm on the quality of the paraphrases in the 128-shot setting across three classification tasks SST2, SST5

Table 1: The average accuracy of the best performing validation checkpoint in the 128-shot SST2 classification task for both the on-policy and KLOn learning techniques. Highest performance per column bolded. Last column reports the macro-average among each row. Numbers in parentheses ($\sigma \mid m$): σ represents the standard deviations for the reported means across five train/validation splits; m reports the average accuracy of all the validation checkpoints as we monitor robustness during the training trajectory. The suffix '-Z' denotes models trained with reward normalization.

Learning Technique	On-Policy Top-P	On-Policy Beam	On-Policy Mixed	KLOn Top-P	KLOn Beam	KLOn Mixed	AVG
No Tuning	67.5	67.5	67.5	67.5	67.5	67.5	67.5
PG	67.9 (2.3 | 53.2)	68.0 (1.4 | 52.0)	67.9 (2.0 | 52.4)	68.5 (1.2 | 67.4)	68.3 (1.9 | 67.4)	69.1 (1.4 | 68.2)	68.3 (1.7 | 60.1)
PG-Z	**71.3** (2.1 | 63.8)	**70.2** (1.7 | 68.6)	**71.2** (1.3 | 66.6)	68.9 (1.3 | 67.6)	68.8 (1.4 | 67.9)	69.8 (1.3 | 68.5)	70.0 (1.5 | 67.2)
MML	69.6 (2.1 | 68.5)	69.1 (1.8 | 67.6)	69.8 (2.0 | 68.6)	**69.5** (2.4 | 68.2)	69.9 (1.9 | 69.0)	70.5 (3.0 | 68.9)	69.7 (2.2 | 68.5)
MML-Z	70.3 (2.7 | 69.1)	**70.2** (2.0 | 68.9)	70.2 (2.3 | 69.1)	68.9 (1.8 | 67.9)	**70.3** (1.7 | 69.0)	**70.6** (2.5 | 68.9)	**70.1** (2.2 | 68.8)

and AGNews. To evaluate the quality of the paraphrases, we report the following five metrics:

Grammar (GR): We evaluate grammar by calculating the perplexity score averaged across the dataset using the GPT-2-Large model (Radford et al., 2019a). A low GR score indicates more grammatical texts.

Lexical Diversity (LD): To assess how paraphrases differ lexically from the *original* input text, we calculate the unigram and bigram Rouge scores between each paraphrase and the original input text (Lin, 2004). We then report $1 - \frac{(Rouge\_1+Rouge\_2)}{2}$ as our lexical diversity metric. A higher LD score indicates greater lexical difference compared to the original text.

Pair-wise Lexical Diversity (PLD): To assess the lexical diversity among the set of *paraphrases* for a given original input text, we calculate LD scores for every pair of paraphrases for an input text, and report the average. A higher PLD score indicates greater diversity among the paraphrases for a specific input text.

Semantic Similarity (SS): To assess how semantically similar paraphrases are to the original input text, we compute the BERTScore's F1 metric (Zhang* et al., 2020) between each paraphrase and the original input text using the BERT-Large model. A higher SS score signifies more semantic similarity with the original text.

Factual Consistency (FC): To measure hallucination in the generated paraphrases with respect to the original input text, we rely on a publicly available factual consistency metric. The model has been trained for textual entailment and summarization datasets with samples annotated for factual consistency[2].

Table 2: Average metrics on the test sets to evaluate the quality of paraphrases across three classification tasks: SST2, SST5, and AGNews datasets. The metrics are further averaged across five training folds for the *FinPara* method. *OrigIn*: Represents the original task inputs. *PrePara*: Corresponds to task inputs obtained from the pre-trained paraphraser. *FinPara*: Indicates task inputs from the finetuned paraphraser in the 128-shot setting. We scale scores into the range of [0-100], except for GR. Better performance per column bolded.

Input Type	GR	LD	PLD	SS	FC
OrigIn	198	n/a	n/a	n/a	n/a
PrePara	**143**	**60.5**	**61.6**	70.4	77.3
FinPara	162	50.6	53.3	**74.7**	**79.1**

We present the metrics for the datasets in Table 2. Compared to a model that was not fine-tuned for this task (PrePara), our RIFF algorithm has reduced the diversity among the generated paraphrases and their lexical variation compared to the original input text. This outcome aligns with our search-learn objective that prioritizes high-scoring paraphrases over others. RIFF has contributed to higher semantic similarity compared to the original input. Interestingly, the perplexity of the generated paraphrases after fine-tuning with our objective is still low, demonstrating the grammatical accuracy of these paraphrases. The example paraphrases shown in Appendix C illustrate that RIFF reduces hallucination in the generated paraphrases, which may contribute to the lower LD score but higher SS score with respect to the original input text. A higher FC score as shown in Table 2 verifies that the RIFF objective has reduced hallucination in the

[2] https://huggingface.co/vectara/hallucination_evaluation_model

generated paraphrases.

3.4 Paraphrases for Few-shot LM Tuning

Our primary hypothesis is that various LM tuning techniques could benefit from diverse views of the original input text. To test this hypothesis, we fine-tuned our paraphrase generators in a 16-shot classification setup using the RIFF algorithm. Subsequently, we fine-tuned the downstream classification model in the same 16-shot setting, while introducing $M = 8$ paraphrases as per the objective outlined in Equation 1. For evaluation, we used the best model from the validation set to make predictions on standard evaluation splits, following the ensemble approach described in Section 2.2. For consistency with prior research, we used the random dataset splits provided by RLPrompt (Deng et al., 2022), aligning with the random seeds used by LM-BFF (Gao et al., 2021).

We study the effect of paraphrases on seven language model tuning techniques. The first technique *AllTune* updates every parameter in the network. Another technique, *GS*, is based on AutoPrompt (Shin et al., 2020) for discrete prompt optimization. The technique *SpTune* (Lester et al., 2021) learns soft prompt vectors, and *LoRA* (Hu et al., 2021) is a recent adaptation technique. Additionally, we investigate *ClsTune*, which trains a classifier on top of the language model, *InTune*, which updates all of the input embedding table, and *HTune*, which only updates the language modeling head in the Transformer architecture. A detailed description of these techniques is discussed in Appendix A.

Table 3 illustrates the average accuracy on standard test sets across six text classification datasets. The reported scores correspond to seven distinct LM tuning techniques: *ClsTune*, *GS*, *SpTune*, *HTune*, *InTune*, *AllTune*, and *LoRA*.

Recent prompt optimization techniques like *GS* and *SpTune* significantly benefit from paraphrase augmentation during training, with *SpTune* demonstrating the most dramatic improvement (2.2% average accuracy increase). While *LoRA* already outperforms these techniques (Hu et al., 2021), paraphrase augmentation further enhances its efficiency in learning adaptation matrices, leading to average accuracy gains of 0.2% on SST2 and 0.3% on AGNews. When coupled with ensemble predictions, denoted in rows with "+RIFF (train+test)", all LM tuning techniques see improvements from the generated paraphrases.

3.5 Paraphrase Robustness Analysis

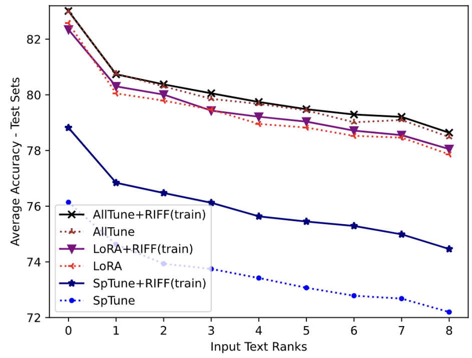

Figure 2: Average test accuracy across six classification datasets. Input at rank 0 represents the original test input, while the remaining eight inputs are top-ranked paraphrases generated by our fine-tuned paraphrase model.

We generate $M = 8$ paraphrases for each test input text and investigate the average performance of the fine-tuned language models on each of these paraphrases compared to the original input text. Figure 2 illustrates the average test accuracy over six classification datasets for *AllTune*, *LoRA*, and *SPTune*, which are popular LM tuning techniques in NLP. The original input texts are denoted with inputs at rank 0, whereas rank_i ($i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$) are the paraphrases returned by the diverse beam search. We observe that the LM tuning techniques are not robust to paraphrases. Our paraphrase augmentation during training increases paraphrase robustness. *SPTune* observes more significant improvement in robustness from paraphrase augmentation during training.

3.6 Limitations Discussion

Our paraphrase generator is pre-trained on semi-supervised paraphrases given by a truly large language model (i.e. ChatGPT). Although these large models are capable of generating high quality paraphrases for the English language. It is not clear if these semi-supervised paraphrases are available for other languages.

In terms of training overhead for our method, once the paraphrase model is fine-tuned, augmenting the mini-batches with paraphrases has minimal impact on training time in the downstream classification task. This is because we only need to perform inference with the paraphrase model once for the training examples to generate their paraphrases before the first epoch. The generated paraphrases are then cached for subsequent training

Table 3: Average accuracy on the standard evaluation sets for the 16-shot text classification. Numbers in parentheses are standard deviations across the five train/validation folds. The last column is the micro averaged performance across the datasets. Highest performance per dataset bolded, second highest underlined. †: the average 16-shot *AllTune* results with automatically searched templates (Gao et al., 2021). ⋆: reported results for RoBERTa-large using in-context learning (Gao et al., 2021).

Tuning Method	SST2	SST5	CR	MR	TREC	AGN	AVG
No Tuning	84.6	31.0	77.8	81.3	27.6	51.5	58.6
ICL⋆	84.8	30.6	87.4	80.5	26.2	-	66.9
RLPrompt	92.5	41.4	89.5	<u>87.1</u>	60.5	80.2	77.7
LM-BFF†	92.3	49.2	89.0	85.5	<u>88.2</u>	-	78.5
ClsTune	72.6 (2.4)	34.4 (2.6)	71.4 (2.7)	67.3 (2.8)	74.8 (3.4)	81.7 (1.6)	70.9 (2.2)
+RIFF (train)	72.5 (3.4)	33.9 (3.7)	68.3 (3.9)	70.3 (0.9)	75.8 (1.7)	84.0 (0.9)	71.9 (2.0)
+RIFF (train+test)	74.0 (3.3)	35.0 (3.6)	71.1 (4.4)	72.0 (1.7)	76.8 (2.9)	84.9 (0.9)	73.3 (2.1)
GS	85.5 (1.3)	37.0 (4.2)	80.2 (2.3)	83.0 (1.8)	45.3 (13.1)	82.0 (1.4)	75.0 (2.3)
+RIFF (train)	86.4 (1.9)	37.8 (3.3)	82.7 (1.3)	84.7 (2.1)	51.0 (8.6)	81.0 (2.4)	75.4 (2.5)
+RIFF (train+test)	87.3 (2.0)	38.2 (3.5)	85.1 (1.9)	84.7 (1.9)	52.4 (7.7)	83.3 (1.5)	77.0 (2.1)
SpTune	89.7 (3.7)	39.4 (6.2)	82.4 (2.8)	86.1 (2.2)	35.2 (2.7)	82.0 (2.6)	76.1 (3.2)
+RIFF (train)	91.2 (2.2)	44.5 (4.2)	84.6 (1.9)	86.1 (0.7)	38.4 (4.3)	84.0 (1.9)	78.3 (2.2)
+RIFF (train+test)	91.6 (2.3)	45.1 (4.1)	86.2 (1.8)	86.6 (0.8)	38.4 (4.0)	86.0 (1.0)	79.7 (1.7)
HTune	87.4 (2.3)	37.4 (2.0)	84.0 (2.8)	83.1 (1.8)	62.4 (7.4)	81.4 (1.4)	76.0 (2.0)
+RIFF (train)	88.1 (1.7)	40.3 (1.9)	84.5 (1.5)	83.4 (2.8)	70.7 (4.8)	83.4 (0.9)	77.8 (1.6)
+RIFF (train+test)	89.1 (1.2)	40.4 (1.8)	86.4 (0.8)	83.1 (3.7)	71.6 (5.9)	85.2 (1.1)	79.0 (1.6)
InTune	91.5 (1.2)	42.3 (4.2)	87.3 (2.0)	84.0 (2.5)	67.7 (5.8)	83.8 (2.2)	78.9 (2.5)
+RIFF (train)	92.6 (0.3)	43.2 (1.9)	87.5 (1.8)	85.9 (2.3)	63.8 (5.3)	85.6 (0.8)	80.2 (1.3)
+RIFF (train+test)	93.1 (0.6)	43.9 (2.3)	89.0 (1.8)	86.0 (2.4)	69.6 (6.3)	86.9 (0.4)	81.3 (1.3)
AllTune	93.1 (0.4)	48.0 (1.0)	89.2 (0.8)	**87.3** (3.1)	87.2 (3.8)	<u>87.7</u> (0.5)	<u>83.0</u> (1.0)
+RIFF (train)	<u>93.6</u> (1.3)	<u>50.6</u> (1.0)	<u>90.2</u> (1.5)	85.8 (2.3)	84.2 (4.9)	87.2 (0.6)	<u>83.0</u> (1.2)
+RIFF (train+test)	**93.8** (1.2)	**51.2** (1.6)	**91.0** (1.6)	85.5 (2.3)	84.4 (4.9)	87.2 (0.6)	**83.2** (1.3)
LoRA	92.5 (1.8)	48.1 (1.8)	88.6 (2.0)	86.0 (2.6)	**89.3** (2.2)	87.3 (0.5)	82.6 (1.3)
+RIFF (train)	92.7 (1.8)	48.0 (2.3)	87.5 (1.5)	85.1 (2.9)	84.8 (2.7)	87.6 (0.3)	82.3 (1.3)
+RIFF (train+test)	93.1 (1.2)	49.2 (2.0)	89.0 (1.1)	85.4 (2.8)	85.9 (3.6)	**87.9** (0.3)	82.9 (1.1)

epochs while fine-tuning the downstream language model.

In contrast, fine-tuning the paraphrase model using the RIFF objective requires generating new paraphrases each epoch, which prevents us from caching paraphrases across epochs. This fine-tuning step is about eight times slower than maximum-likelihood training, which benefits from having ground-truth paraphrases available. Generating new samples and scoring them is a standard procedure in Reinforcement Learning.

4 Related Works

To improve prompt optimization and efficient tuning techniques for LMs, we incorporate the generated paraphrases into the training mini-batches. Paraphrase generation represents just one technique of data augmentation. For a comprehensive overview of diverse data augmentation techniques for NLP tasks, we direct interested readers to a recent survey by Chen et al. (2021).

A recent work for few-shot prompt-based learning helps contrastive training by paraphrasing the inputs (Abaskohi et al., 2023). Our work proposes an objective to further fine-tune the paraphrase generator distilled from an LLM that reduces hallucination. Despite the previous work, which only studies the *AllTune* technique, we investigate the impact of paraphrases for various language model tuning techniques. Without contrastive learning, we can show that we can improve LM's robustness by paraphrase generation during training.

Prompt Optimization & Efficient Tuning: Recent research proposes various techniques for prompt optimization and efficient tuning. We have used successful techniques from each of these areas. Appendix E provides our detailed description of these recent techniques. All of the recent techniques for prompt optimization and efficient tuning use the original input context provided within the dataset.

Paraphrase Generation: Recent techniques encompass various approaches, including the use of copy mechanisms, Variational Autoencoders, Generative Adversarial Networks, and Reinforcement Learning techniques to generate diverse paraphrases (Zhou and Bhat, 2021). While previous studies have applied RL techniques for paraphrase generation, we propose the use of MML gradients instead of policy gradients to fine-tune our paraphrase model by the reward of a secondary classification task (see Appendix E for more discussion).

5 Conclusion

We investigated the impact of incorporating input paraphrases while fine-tuning PLMs with recent efficient tuning techniques. Our results indicate that specific techniques, such as continuous and discrete prompt optimization methods like AutoPrompt or Soft-Prompt Tuning, benefit significantly from the inclusion of paraphrases. We also conducted extensive experiments to reduce noise in a distantly supervised paraphrase generator. Our ablation studies on fine-tuning the paraphrase generator demonstrate that policy gradient objectives lack robustness during training, while maximum marginal likelihood training remains a robust technique.

Ethics Statement

Many language models show biases in their output due to the data used to train them (Liang et al., 2021). It is possible that even with few-shot language model tuning, we might continue to detect analogous biases in the downstream classification task, for instance, resulting in diminished classification accuracy for specific minority groups. It is also possible that the additional data generated by the paraphrase model will exaggerate existing biases.

Acknowledgements

We gratefully acknowledge the Vector Institute, and the Digital Research Alliance of Canada (alliancecan.ca) for providing access to their GPU cluster, which streamlined our experiments. This research was funded by NSERC, Alberta Machine Intelligence Institute (Amii), and CIFAR. Alona Fyshe is a Canada CIFAR AI Chair.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El

Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

David Broughton. 1995. *The assumptions and theory of public opinion polling*, pages 15–33. Macmillan Education UK, London.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in NLP. *CoRR*, abs/2106.07499.

Ganqu Cui, Wentao Li, Ning Ding, Longtao Huang, Zhiyuan Liu, and Maosong Sun. 2023. Decoder tuning: Efficient language understanding as decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15072–15087, Toronto, Canada. Association for Computational Linguistics.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Margaret C. Donaldson. 1978. *Children's Minds*.

Wanyu Du and Yangfeng Ji. 2019. An empirical comparison on imitation learning and reinforcement learning for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6012–6018, Hong Kong, China. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Sonal Garg, Sumanth Prabhu, Hemant Misra, and G. Srinivasaraghavan. 2021. Unsupervised contextual paraphrase generation using lexical control and reinforcement learning. *CoRR*, abs/2103.12777.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2022. Efficient (soft) Q-learning for text generation with limited good data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6969–6991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2021. Efficient (soft) q-learning for text generation with limited good data.

Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1051–1062, Vancouver, Canada. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019a. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019b. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2019. Averaging weights leads to wider optima and better generalization.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards Understanding and Mitigating Social Biases in Language Models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2020. Exploring versatile generative language model via parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 441–459, Online. Association for Computational Linguistics.

Mingtong Liu, Erguang Yang, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2020. A learning-exploring method to generate diverse paraphrases with multi-objective deep reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2310–2321, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783.

Saeed Najafi and Alona Fyshe. 2023. Weakly-supervised questions for zero-shot relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3075–3087, Dubrovnik, Croatia. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3173–3182, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019b. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2019. On the convergence of adam and beyond. *CoRR*, abs/1904.09237.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563.

Abhilasha Sancheti, Balaji Vasan Srinivasan, and Rachel Rudinger. 2022. Entailment relation aware paraphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11258–11266.

Weijia Shi, Xiaochuang Han, Hila Gonen, Ari Holtzman, Yulia Tsvetkov, and Luke Zettlemoyer. 2022. Toward human readable prompt tuning: Kubrick's the shining is a good movie, and a good prompt too?

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

A. B. Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1800–1809, New York, NY, USA. Association for Computing Machinery.

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

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. *CoRR*, abs/2201.03514.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA. MIT Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, Dubrovnik, Croatia. Association for Computational Linguistics.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Maxim Kuznetsov Vladimir Vorobev. 2023. Chatgpt paraphrases dataset.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2022b. Tempera: Test-time prompting via reinforcement learning.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022c. Automatic chain of thought prompting in large language models.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers.

A Baseline LM Tuning Techniques

Gradient-Search (*GS*): The *GS* technique is based on the recent AutoPrompt (Shin et al., 2020) method, which optimizes task instructions without updating any parameters in the model. The search process begins in the vocabulary space, optimizing the change in label log-likelihood when replacing token p_i in the task instruction with another token v from the vocabulary set. In our implementation, each search iteration randomly selects one mini-batch of training examples and then randomly selects a token from the task instruction to update. The top k candidate tokens are determined based on the approximate change in label log-likelihood: $Top_v \ \{w_v^T . \nabla_{w_{p_i}} \log P_{\text{lm}}(y|p,x)\}$, where w_v is the embedding vector of a candidate token v. The resulting k new task instructions are evaluated again using label log-likelihood on the same training examples[3], and the top-performing instruction is retained for the next search iteration. Prompt optimization always uses the original input x when searching new task prompts (Shin et al., 2020; Deng et al., 2022). In our work, we investigate the impact of incorporating paraphrases of x during search.

Input-Finetuning (*InTune*): As a straightforward and efficient tuning technique, we compare to updating only the input embedding table in the transformer architecture. This method requires gradient computation similar to All-Finetuning (*AllTune*) as well as the *GS* method.

LM-Head-Finetuning (*HTune*): The transformer-based pre-trained language models consist of a language modeling head, which maps the hidden vectors to the token logit for each token in the vocabulary. For the *HTune* technique, we solely update the parameters of the language modeling head.

Classifier-Finetuning (*ClsTune*): In *ClsTune*, we first create a feature representation $h(x)$ for the input text x using average pooling of the final hidden vectors in the last layer of the language model. Here, we assume that the language model (feature extractor) remains fixed, and we then construct a two-layer feedforward network with the $gelu$ activation function (Hendrycks and Gimpel, 2016) as a classification module on top of the language model.

Softprompt-Tuning (*SpTune*): In *SpTune* (Lester et al., 2021), L prompt tokens are prepended to the task instruction. These L tokens are associated with L dedicated prompt embedding vectors, extending the sequence of vectors derived from the task instruction and input text with an additional L trainable feature vectors. During training, the original embedding table of the transformer model remains fixed, while a new prompt embedding table is trained by backpropagating the label log-likelihood into the prompt embedding table. In contrast to *InTune*, here the prompt vectors do not need to map to vocabulary words.

Low-Rank Adaptation (*LoRA*): *LoRA* is one of the latest efficient-tuning techniques specifically designed for PLMs (Hu et al., 2021). It learns low-rank adaptation matrices for the query and value weight matrices within the transformer model. For a pre-trained weight matrix $W_q \in \mathbb{R}^{d \times k}$, *LoRA* learns the necessary adaptation (i.e., modification) of the weight matrix for a downstream task through a low-rank decomposition, expressed as $W_q + \triangle W_q \approx W_q + BA$. Here, $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \leq min(d,k)$. The adaptation matrices A and B are the only parameters subject to training, while the original matrix W_q does not receive any gradient updates. Studies have shown that *LoRA* performs on par with, or better than, *AllTune* across various PLMs (Hu et al., 2021).

All language model tuning techniques we have discussed will use the same input format. For example in the sentiment classification task, we use the following format:

"<s> {instruction} {text} . It was <mask> . </s>".

Except for *ClsTune*, all of our tuning techniques maximize the probability of the correct label token in place of the <mask> token. In contrast, *ClsTune* takes the formatted input and classifies it into one of the predefined class labels.

B Few-shot Paraphrase Fine-Tuning (Further Results)

This section provides additional results that compare our training objectives for fine-tuning the paraphrase generator using the feedback from the downstream language model.

The off-policy learning technique improves performance when using basic rewards (i.e., 69.1% compared to 67.9% with mixed decoding). However, the combined effect of off-policy learning and reward normalization decreases performance. With mixed decoding, 'PG-Z' yields an accuracy of 71.2% in on-policy learning compared to an accuracy of 68.0% with off-policy learning. The 'AVG' column in Table 4 further verifies this conclusion that reward normalization is not improving the final performance while training the model with off-policy learning. We hypothesize that with the off-policy learning technique, the normalized rewards should be re-weighted properly if the sampled para-

[3] The original AutoPrompt evaluates the new candidate instructions on another training mini-batch. For fewshot classification, we re-use the drawn training mini-batch to evaluate the complete new candidate instructions.

Table 4: The accuracy of the best performing validation checkpoint in the 128-shot SST2 classification task trained with the off-policy learning technique.

	Off-Policy			AVG
Learn Tech	Top-P	Beam	Mixed	
No Tuning	67.5	67.5	67.5	67.5
PG	68.6 (1.8 | 67.5)	68.4 (1.6 | 67.4)	69.1 (1.6 | 67.5)	68.7 (1.7 | 67.5)
PG-Z	68.8 (1.7 | 67.5)	68.7 (1.1 | 67.4)	68.0 (2.1 | 67.2)	68.5 (1.6 | 67.4)
MML	**69.2** (2.8 | 68.0)	**70.1** (2.4 | 68.6)	**70.1** (3.3 | 68.4)	**69.8** (2.7 | 68.3)
MML-Z	**69.2** (2.5 | 68.3)	69.7 (3.5 | 68.6)	**70.1** (2.8 | 68.6)	69.7 (2.9 | 68.5)

phrases are from the fixed paraphrase model.

C Example Paraphrases

In Table 5 and Table 7, we present example paraphrases from the AGNews test dataset generated by our pre-trained paraphrase model. We selected the AGNews dataset for its suitability in paraphrasing longer texts or short paragraphs. Subsequently, in Table 6 and Table 8, we display the generated paraphrases after fine-tuning the paraphrase model with the RIFF objective. Fewer hallucinations can be observed in the new paraphrases, which are highlighted in red.

D Further Training Details

The learning rate for each LM tuning technique was separately fine-tuned from the set {0.5, 0.3, 0.1, 0.01, 0.001, 0.0001, 0.00001} using the train/validation split created for the seed 11 on the SST2 dataset. The tuned learning rates were then applied globally across other datasets and experiments. For paraphrase fine-tuning, we train all the parameters in T5-base with the learning rate of 0.00001. In Tables 9 and 10, we list the hyper-parameters and learning rates used across all datasets. For optimization, we utilized the AdamW (Loshchilov and Hutter, 2017)[4] optimizer with the AMSGrad variant set to True (Reddi et al., 2019). We implemented the methods using the HuggingFace[5] library and the PyTorch[6] machine learning framework. We report the accuracy metric on these classification datasets. The experiments were conducted using multiple NVIDIA's A40 GPU cards.

[4] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html
[5] https://huggingface.co/
[6] https://pytorch.org/

E Extended Related Works

Prompt Optimization & Efficient Tuning: Recent research proposes various techniques for prompt optimization and efficient tuning of language models. In our experiments, we have used successful techniques from each of these areas.

FluentPrompt (Shi et al., 2022) is a recent discrete prompting technique based on the projected gradient-descent and Langevin dynamics. FluentPrompt introduces a fluency constraint within Langevin dynamics to generate a sample of high-performing prompts for more interpretable analysis of these discrete prompts. The optimized prompts by FluentPrompt performs on-par to the AutoPrompt, however they have lower perplexity (Shi et al., 2022).

Building upon *SpTune* (Lester et al., 2021) and P-tuning (Li and Liang, 2021), P-tuning V2 (Liu et al., 2022) introduced the concept of deep prompt tuning. This method involves injecting prompt vectors into the deeper layers of the transformer model to close the performance gap with AllTuning in medium-sized language models. We have experimented with *LoRA* (Hu et al., 2021), a recent low-rank adaptation technique for tuning language models. Other potential methods include training bottleneck adapter modules (Houlsby et al., 2019b; Lin et al., 2020) added per sub-layer of the transformer model. *LoRA* outperforms adapter tuning and P-Tuning V2 techniques (Hu et al., 2021). The successors of *LoRA* include DyLoRA (Valipour et al., 2023) which dynamically learns a range of adaptation ranks, thus eliminating the need to search the rank of the adaptation matrices as a hyper-parameter. Similarly, Ada*LoRA* dynamically allocates the parameter budget among the weight matrices during adaptation, with matrices of higher priority (i.e., those with greater importance to the downstream task) receiving higher adaptation ranks than less important matrices (Zhang et al., 2023).

Table 5: Generated top eight paraphrases using diverse beam search **with the pre-trained paraphrase model (distilled from GPT3.5-turbo)** for the following input sentence from the AGNews test dataset: "*Gardner Loses Quest for Repeat Wrestling Gold US heavyweight Rulon Gardner lost his Olympic title Wednesday after being beaten in the semi-final stage of the 120kg Greco-Roman wrestling event by Georgiy Tsurtsumia of Kazakhstan.*" Contradicting information or hallucinations are highlighted in red.

Paraphrase	Text
P1	"*On Wednesday, US heavyweight Rulon Gardner lost his Olympic title after losing in the semi-finals of the 120kg Greco-Roman wrestling event against Georgiy Tsurtsumia of Kazakhstan.*"
P2	"*Following his loss to Georgiy Tsurtsumia of Kazakhstan in the semi-finals of the 120kg Greco-Roman wrestling event, US heavyweight Rulon Gardner lost his bid for repeat Wrestling Gold.*"
P3	"*Rulon Gardner, the heavyweight world champion, suffered a major upset on Wednesday as he was defeated in the semi-finals of the 120kg Greco-Roman wrestling tournament by Georgiy Tsurtsumia of Kazakhstan. Gardner lost his Olympic title bid to George Kubica after losing 1-1 and 6-1.*"
P4	"*US heavyweight Rulon Gardner suffered a setback in the final round of his Olympic title against Georgiy Tsurtsumia of Kazakhstan, leading to an upset loss that could have resulted in injury.*"
P5	"*Despite winning the Olympic title in 2004, Rulon Gardner, US heavyweight champion, lost his quest to repeat as a wrestler in the 120kg Greco-Roman wrestling event after losing to Georgiy Tsurtsumia of Kazakhstan in their semi-final match.*"
P6	"*The US heavyweight champion, Rulon Gardner of the United States, suffered a defeat to Georgiy Tsurtsumia from Kazakhstan in the semi-finals of his men's 120kg Greco-Roman wrestling tournament, which cost him the Olympic championship on Wednesday.*"
P7	"*In a bid to avoid repeat Wrestling Gold, US heavyweight Rulon Gardner lost his Olympic title on Wednesday after losing in the final round of Georges Tsurtsumia'S 120kg Greco-Roman wrestling match against Kazakhstan's Rorum Ignite.*"
P8	"*After losing his Olympic title on Wednesday, US heavyweight Rulon Gardner was defeated in the semi-finals of Greco–Roman wrestling event at 120kg against Georgiy Tsurtsumoj, Kazakhstan.*"

Table 6: Generated top eight paraphrases using diverse beam search **after fine-tuning the paraphrase model with the RIFF method** in the 128-shot setting. The example input sentence is from the AGNews test dataset: "*Gardner Loses Quest for Repeat Wrestling Gold US heavyweight Rulon Gardner lost his Olympic title Wednesday after being beaten in the semi-final stage of the 120kg Greco-Roman wrestling event by Georgiy Tsurtsumia of Kazakhstan.*" Contradicting information or hallucinations are highlighted in red

Paraphrase	Text
P1	"*The US heavyweight Rulon Gardner lost his Olympic title on Wednesday after losing in the semi-final stage of the 120kg Greco-Roman wrestling event to Georgiy Tsurtsumia of Kazakhstan.*"
P2	"*On Wednesday, Rulon Gardner lost his Olympic title after losing in the semi-final stage of the 120kg Greco-Roman wrestling event to Georgiy Tsurtsumia of Kazakhstan.*"
P3	"*Georgiy Tsurtsumia of Kazakhstan defeated US heavyweight Rulon Gardner in the semi-final stage of the 120kg Greco-Roman wrestling event, resulting in Gardner losing his Olympic title on Wednesday.*"
P4	"*Despite losing his Olympic title on Wednesday, US heavyweight Rulon Gardner lost his quest for repeat wrestling. He was defeated in the 120kg Greco-Roman wrestling event by Georgiy Tsurtsumia of Kazakhstan in their semi-final stage.*"
P5	"*Rulon Gardner, the heavyweight US heavy weight loser of Olympic title after losing in the semi-final stage of the 120kg Greco-Roman wrestling event to Georgiy Tsurtsumia of Kazakhstan.*"
P6	"*US heavyweight Rulon Gardner suffered a loss in the semi-final stage of the 120kg Greco-Roman wrestling event after losing his Olympic title.*"
P7	"*In the 120kg Greco-Roman wrestling event, Rulon Gardner lost his Olympic title on Wednesday after losing in the semi-final stage by Georgiy Tsurtsuma of Kazakhstan.*"
P8	"*After losing his Olympic title on Wednesday, Rulon Gardner lost the semi-final stage of a 120kg Greco–Roman wrestling event against Georgiy Tsurtsumoia of Kazakhstan in the Quest for Repeat Wrestling.*"

Table 7: Generated top eight paraphrases using diverse beam search **with the pre-trained paraphrase model (distilled from GPT3.5-turbo)** for the following input sentence from the AGNews test dataset:
"*Calif. Aims to Limit Farm-Related Smog (AP) AP - Southern California's smog-fighting agency went after emissions of the bovine variety Friday, adopting the nation's first rules to reduce air pollution from dairy cow manure.*"
Contradicting information or hallucinations are highlighted in red.

Paraphrase	Text
P1	"*The smog-fighting agency in southern California followed the lead of other states by implementing the first rules to control air pollution from dairy cow manure emissions.*"
P2	"*Southern California's smog-fighting agency took action on bovine eutrophication by adopting the first rules of the nation to reduce farm-related air pollution, including dairy cow manure emissions.*"
P3	"*California's smog-fighting agency in southern California took action on bovine deposition Friday, following the implementation of the first rules in the country to control air pollution from dairy cow manure.*"
P4	"*Calf Air Pollution Control: Southern California's smog-fighting agency took action on Friday to limit farm-related sewage emissions by adopting the first rules in the country to reduce air pollution from dairy cow manure.*"
P5	"*Efforts to Reduce Farm-Related Smog: Calif.'s agency in Southern California set the pace on Friday with its first-ever goals to combat bovine emissions and the nation'll be moving quickly to eliminate farm-related air pollution, including dairy cow manure.*"
P6	"*Southern California's EPA sought to reduce air pollution from farm-related manure, and the agency adopted the first rules in history to address bovine smog emissions.*"
P7	"*Calif. aims to reduce farm-related smog, which Southern Californiaś first specialized agency for reducing bovine emissions took action on Friday by adopting the nation' reviews that address this issue with ""first ever"" rules from the U.S. federal government.*"
P8	"*The smog-fighting agency in Southern California adopted new regulations on Friday to reduce farm-related pollution, including air pollution from dairy cow manure.*"

Table 8: Generated top eight paraphrases using diverse beam search **after fine-tuning the paraphrase model with the RIFF method** in the 128-shot setting. The example input sentence is from the AGNews test dataset:
"*Calif. Aims to Limit Farm-Related Smog (AP) AP - Southern California's smog-fighting agency went after emissions of the bovine variety Friday, adopting the nation's first rules to reduce air pollution from dairy cow manure.*"
Contradicting information or hallucinations are highlighted in red

Paraphrase	Text
P1	"*The smog-fighting agency in southern California adopted the first rules to reduce air pollution from dairy cow manure, targeting bovine emissions.*"
P2	"*Southern California's smog-fighting agency took action on bovine emissions by adopting the nation' first rules to reduce air pollution from dairy cow manure.*"
P3	"*On Friday, Southern California's smog-fighting agency adopted the nation' first rules to reduce air pollution from dairy cow manure emissions.*"
P4	"*California's smog-fighting agency took action on bovine emissions by adopting the nation' first rules to reduce air pollution from dairy cow manure.*"
P5	"*Calif. seeks to reduce farm-related smog emissions by adopting the nation's first rules on Friday, which targets bovine emission in Southern California and is one of the country' best agencies for curbing air pollution from dairy cow manure.*"
P6	"*Southern California's scounding agency took action on bovine emissions by adopting the nation' first rules to reduce air pollution from dairy cow manure.*"
P7	"*The smog-fighting agency of Southern California targeted bovine emission on Friday, adoptting the country's first rules to reduce air pollution from dairy cow manure.*"
P8	"*AP: Southern California's federal agency adopting the nation' first rules to reduce air pollution from dairy cow manure.*"

Table 9: Shared hyper-parameters used across all experiments and datasets.

Hyper-parameter	Value
Top-k candidates in *GS*	k=4
batch size (RoBERTa-large)	8
batch size in *GS* (RoBERTa-large)	2
Weight decay	0.0001
Max epochs	100
length cutoff	128 tokens
Paraphrase sample size	M=8
Checkpointing steps	8
$D^{'}$ in *ClsTune*	128
Prompt len in *SpTune*	L=25
β in MML	0.1
β in PG	0.6
LoRA α	32
LoRA r	8
LoRA dropout	0.1
Diversity penalty for Div beam	3.0
Repetition penalty for Div beam	10.0
Temperature in Div beam	0.7
P value for top-p	0.99

Table 10: Learning rates used per Language Model (LM) tuning technique.

LM Tuning Technique	Learning Rate
GS	No rate
AllTune	0.00001
InTune	0.001
HTune	0.001
ClsTune	0.001
SpTune	0.001
LoRA	0.0001

In scenarios where gradients are absent, Black-Box Tuning (Sun et al., 2022) applies derivative-free algorithms for optimizing continuous prompts. For discrete prompt optimization, RLPrompt (Deng et al., 2022) employs the on-policy version of soft Q-learning (Guo et al., 2021) to find the optimal prompt tokens in a gradient-free setting. Decoder Tuning (Cui et al., 2023) learns a decoder network over the language model, thus circumventing the need for gradient computation and input-side prompt tuning in few-shot classification. In a recent study, TEMPERA (Zhang et al., 2022b) introduced a novel approach that involves test-time discrete prompt editing using a trained RL agent. This agent is capable of modifying the instruction, in-context examples, or the verbalizers based on the given task input.

The use of Language Models (LLMs) in generating instructions for downstream tasks has involved a two-step process. Initially, LLMs generate a set of candidate instructions, and subsequently, the highest-scoring instruction is utilized to prompt another LLM to perform the downstream task. This approach, known as prompt-based generation-then-filtering, has been investigated in the recent APE method (Zhou et al., 2023). APE demonstrates the ability to generate prompts that achieve performance comparable to human-designed prompts (Zhou et al., 2023).

To prompt language models for reasoning tasks, another line of research augment the input context with demonstration examples outlining the intermediate reasoning steps to form the answer. Providing manually or automatically generated chain-of-thoughts within these demonstrations strikingly improve LLMs performance in reasoning tasks (Wei et al., 2022; Zhang et al., 2022c; Kojima et al., 2022).

All of the aforementioned techniques for prompt optimization and efficient tuning of the language model use the original task's input text (or the original input context) provided within the dataset.

RL for Paraphrase Generation: In the following paragraphs, we provide a brief overview of similar reinforcement learning objectives employed for paraphrase generation. Li et al. (Li et al., 2018) used a deep RL technique, training a pointer-generator network as the paraphrase generator and a decomposable attention model as the evaluator which assigns a paraphrase score to pairs of sentences. The generator was trained using the policy gradient objective, with reward shaping and scaling to stabilize the training process (Li et al., 2018). Another approach by Qian et al. (Qian et al., 2019) focused on generating diverse paraphrases by training multiple generators, accompanied by a paraphrase discriminator and a generator discriminator. Policy gradient objective and self-critical learning (Rennie et al., 2016) were employed for training the generators, with the baseline reward used in the policy gradient objective being the reward obtained from the greedy-decoded sequence. Liu et al. (Liu et al., 2020) also applied the policy gradient objective with self-critical learning, incorporating multiple reward functions such as Rouge score with the reference paraphrase, negative Rouge score with the input sentence to encourage lexical variations, and semantic similarity score between the paraphrase and the input sentence to ensure semantic fidelity.

Another study by Du and Ji (Du and Ji, 2019) compared the use of imitation learning algorithm DAGGER with policy gradient REINFORCE for paraphrase generation. The policy gradient objective has also been applied in generating paraphrases while considering multiple objectives for entailment relation-aware paraphrase generation (Sancheti et al., 2022). In the context of chatbot responses, a recent work studies unsupervised paraphrase generation with proximal policy optimization, aiming to maximize a combination of rewards such as textual entailment, semantic similarity, language fluency, and lexical dissimilarity (Garg et al., 2021). Similarly, the policy gradient objective has been employed to optimize multiple rewards, similar to previous work, for unsupervised paraphrase generation (Siddique et al., 2020).

While previous studies have applied RL techniques for paraphrase generation, we propose the use of MML gradients instead of policy gradients to train our paraphrase model. Our training objective fine-tunes the paraphrase model for a downstream classification task. Our paraphrase model has been distilled from a large language model.

F Task Instructions & Input Format

Table 11 provides a summary of the task instructions that we append before the inputs, as well as the class verbalizers for classifying the input text. The instructions and input templates are derived from prior work in prompt optimization (Deng et al., 2022).

Table 11: Number of classes C, test set size T, the input format, and the instruction used per dataset. The label words are provided within the instructions.

Dataset	C	T	Input Format	Instruction
SST2	2	1821	"<s> {Instruction} {Text} . It was <mask> . </s>"	"In this task, you are given sentences from movie reviews. The task is to classify a sentence as 'great' if the sentiment of the sentence is positive or as 'terrible' if the sentiment of the sentence is negative."
SST5	5	2210	"<s> {Instruction} {Text} . It was <mask> . </s>"	"In this task, you are given sentences from movie reviews. Based on the given review, classify it to one of the five classes: (1) terrible, (2) bad, (3) okay, (4) good, and (5) great."
CR	2	2000	"<s> {Instruction} {Text} . It was <mask> . </s>"	"In this task, you are given sentences from customer reviews. The task is to classify a sentence as 'great' if the sentiment of the sentence is positive or as 'terrible' if the sentiment of the sentence is negative."
MR	2	2000	"<s> {Instruction} {Text} . It was <mask> . </s>"	"In this task, you are given sentences from movie reviews. The task is to classify a sentence as 'great' if the sentiment of the sentence is positive or as 'terrible' if the sentiment of the sentence is negative."
TREC	6	500	"<s> {Instruction} <mask>: {Text} . </s>"	"You are given a question. You need to detect which category better describes the question. Answer with 'Description', 'Entity', 'Expression', 'Human', 'Location', and 'Number'."
AG's News	4	7600	"<s> {Instruction} <mask> News: {Text} . </s>"	"In this task, you are given a news article. Your task is to classify the article to one out of the four topics 'World', 'Sports', 'Business', 'Tech' if the article's main topic is relevant to the world, sports, business, and technology, correspondingly. If you are not sure about the topic, choose the closest option."