Monotonic Paraphrasing
Improves Generalization of Language Model Prompting

Qin Liu **Fei Wang** **Nan Xu** **Tianyi Yan** **Tao Meng** **Muhao Chen**
UC Davis; USC; UCLA
{qinli, muhchen}@ucdavis.edu;
{fwang598, nanx, tianyiy}@usc.edu; tmeng@cs.ucla.edu

Abstract

Performance of large language models (LLMs) may vary with different prompts or instructions for even the same task. One commonly recognized factor for this phenomenon is the model's familiarity with the given prompt or instruction, which is typically estimated by its perplexity. However, finding the prompt with the lowest perplexity is challenging, given the enormous space of possible prompting phrases. In this paper, we propose monotonic paraphrasing (MONOPARA), an end-to-end decoding strategy that paraphrases given prompts or instructions into their lower perplexity counterparts based on an ensemble of a paraphrase LM for prompt (or instruction) rewriting, and a target LM (i.e. the prompt or instruction executor) that constrains the generation for lower perplexity. The ensemble decoding process can efficiently paraphrase the original prompt without altering its semantic meaning, while monotonically decreasing the perplexity of each generation as calculated by the target LM. We explore in detail both greedy and search-based decoding as two alternative decoding schemes of MONOPARA. Notably, MONOPARA does not require any training and can monotonically lower the perplexity of the paraphrased prompt or instruction, leading to improved performance of zero-shot LM prompting as evaluated on a wide selection of tasks. In addition, MONOPARA is also shown to effectively improve LMs' generalization on perturbed and unseen task instructions.[1]

1 Introduction

Large language models (LLMs) have demonstrated remarkable proficiency in zero-shot decision making (Gonen et al., 2023; Schick and Schütze, 2021; Brown et al., 2020) and instruction following (Jiang et al., 2023; Köpf et al., 2023; Touvron et al., 2023b; Taori et al., 2023; Chiang et al., 2023; Ouyang et al., 2022). However, there can be significant variance in the performance of seemingly similar prompts (Zhao et al., 2021; Lu et al., 2022c; Webson and Pavlick, 2022; Gonen et al., 2023; Yan et al., 2024). Despite efforts of studies on prompting LMs (Shin et al., 2020; Li and Liang, 2021; Gao et al., 2021; Ding et al., 2022; Sanh et al., 2021; Kojima et al., 2022), it is still challenging to develop high-quality prompts that can induce better performance for varying tasks on evolving models in an effort-saving manner.

One consensus reached by recent studies is the inverse relationship between a prompt's perplexity and its task performance (Kumar et al., 2023; Gonen et al., 2023). This stems from the intuition that the frequency of a prompt (or an instruction)[2] appearing in the (pre-)training data positively influences the model's familiarity with it, thereby enhancing its capability to perform the described task (Iter et al., 2023; Gonen et al., 2023; Wang et al., 2023; Lee et al., 2023). As a result, existing attempts use specific criteria, especially perplexity for selecting prompts from a collection of candidates. For example, Gonen et al. (2023) builds a large prompt pool for each task and selects the one with the lowest perplexity. However, the performance of such *rewrite-then-select* methods (Jiang et al., 2020; Zhou et al., 2022; Gonen et al., 2023; Prasad et al., 2023) is limited by the size, quality, and even availability of the candidate pool. Searching the prompt with the lowest perplexity for a particular task remains challenging due to the vast expanse of potential prompts.

To address this challenge, we propose a novel progressive *end-to-end* prompt refining approach, namely monotonic paraphrasing (MONOPARA), that can *proactively* rewrite the given prompt into its low-perplexity counterpart without compromis-

[1]Our code is available at https://github.com/luka-group/MonoPara.

[2]We use the terms "prompt" and "instruction" interchangeably in this paper as they both refer to a natural language command for zero-shot inference.

Findings of the Association for Computational Linguistics: EACL 2024, pages 9861–9877

ing its expressiveness of the task. During prompt (or instruction) refinement, MONOPARA conducts an ensemble decoding process of a paraphrasing model together with the target LM. The paraphrase model is instructed to rewrite the given prompt of a specific task, while the target model, which is later on used for task inference, provides a constraint that aims to lower the perplexity of the generation. MONOPARA iteratively decodes tokens based on the ensemble of these two models. Intuitively, the paraphrase model can genuinely paraphrase the original prompt without altering its semantic meaning and remain instructive for the given task, while the target model restricts the search space of paraphrased tokens to those with low perplexity, resulting in a task prompt that the target model is more familiar with.

Further, for the decoding process of MONOPARA, we explore two decoding schemes. In addition to *greedy decoding* (§2.3) that iteratively generates the next token based on the weighted combination of predicted probabilities from both models, we also explored *search-based decoding* (§2.4). The search-based decoding scheme follows the look-ahead decoding paradigm (Lu et al., 2022b), which keeps several sequence candidates and maintains the one with the lowest perplexity scored by the target model. Compared to recent prompt refinement approaches (Gonen et al., 2023; Kumar et al., 2023; Shum et al., 2023), MONOPARA needs zero training effort and far less computational cost, and do not require the pre-existence of multiple prompt or instruction candidates (Gonen et al., 2023). We find that by an ensemble of two LMs, we are able to consistently decode lower perplexity prompts, which would, in turn, bring about better generalization of LMs on downstream tasks.

Our contributions are three-fold. First, we propose MONOPARA, a prompt refinement method that paraphrases prompts to be more familiar with the target model for boosted generalization and task performance. Second, we explore and compare two distinct decoding strategies that proactively refine prompts by monotonically lowering their perplexity and maintaining their expressiveness of the task. Third, we conduct experimentation for both prompt refinement for regular LMs and instruction refinement for instruction-tuned LMs to illustrate that monotonic paraphrasing improves the generalization of LM prompting.

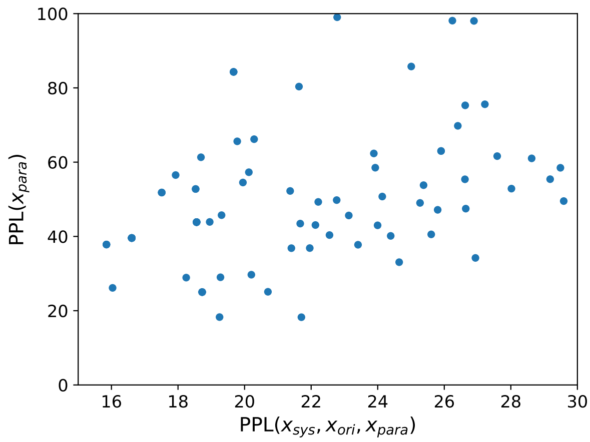

Figure 1: Perplexity of x_{para} as the output paraphrase of P_{para} vs. as the input prompt of P_{tar} for the AG News dataset with Mistral 7B as both P_{para} and P_{tar}. Each point stands for a different prompt x_{para}. A low-perplexity paraphrase does not necessarily result in a low-perplexity prompt for the target model.

2 Monotonic Paraphrasing

We propose MONOPARA as an ensemble-based decoding method that can refine a given prompt by paraphrasing it into a low-perplexity counterpart. We first provide the intuition of MONOPARA in §2.1 and the basics of prompt paraphrasing in §2.2. Then, we formally define two decoding schemes of MONOPARA in §2.3 and §2.4.

2.1 Intuition and Methodology

Assume that the user wants to address a task specified by a given prompt x_{ori} using a target LM P_{tar}. MONOPARA seeks to refine the given prompt to its low-perplexity counterpart x_{para} (i.e. a more *familiar* prompt to P_{tar}), thereby enhancing model performance. However, simply paraphrasing the original prompt with a paraphrase model P_{para} does not necessarily result in a low-perplexity counterpart. This discrepancy arises from a mismatch between the perplexity of the output x_{para} from the paraphrase model and the perplexity observed when using x_{para} as the input prompt for the target model. As shown in Fig. 1, there is a lack of significant correlation between the perplexity of x_{para} as output in P_{para} and that of x_{para} as input in P_{tar}, even if we use the same model as the paraphrase and target model ($P_{para} = P_{tar}$). This observation indicates that the minimal perplexity of a paraphrased prompt could not be achieved by the paraphrase model alone.

Based on the above fact, the goal of MONOPARA is to take on the semantic hints from the paraphrase

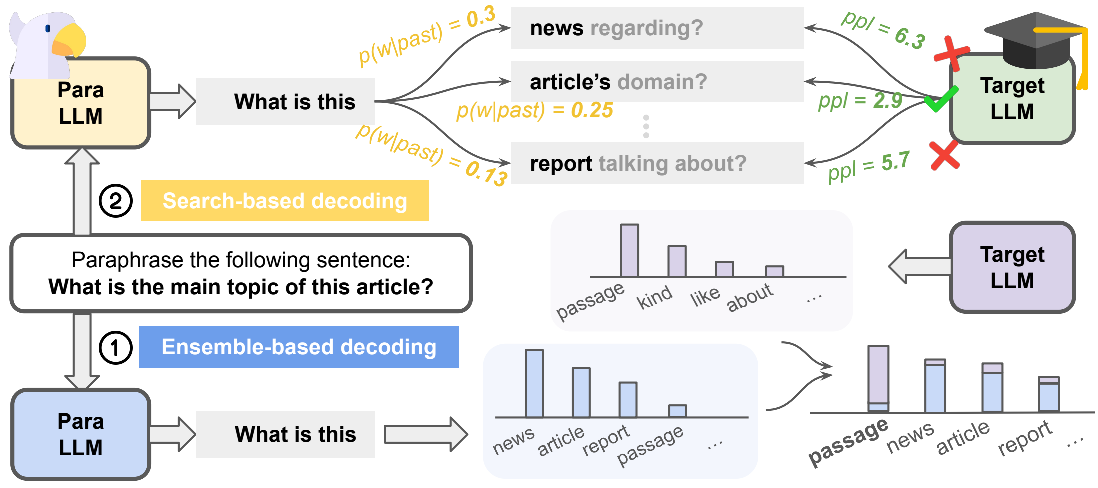

Figure 2: Two explored decoding schemes of MONOPARA. Ensemble-based decoding (bottom) combines the token probabilities from the paraphrase model and the target model in each decoding step. Search-based decoding (top) further leverages look-ahead decoding to consider the potential future impact of current choices.

model while following the constraint of perplexity from the target model. Specifically, the paraphrase model generates paraphrases for a given prompt, while the target model constrains the paraphrase perplexity during the generation process. With an ensemble of these two models, we can leverage the strengths of both: the guided content generation capability of the paraphrase model and the low-perplexity prompt constraint of the target model. MONOPARA is thus able to produce paraphrases with lower perplexity by synergizing the benefits of both models.

2.2 Prompt Paraphrasing

We first introduce prompt paraphrasing without perplexity constraint. We consider using an LLM for prompt paraphrasing in a zero-shot manner, where the model is instructed to paraphrase the original prompt x_{ori} to a fluent and coherent counterpart x_{para} with a *system prompt* x_{sys}, such as the following one:

The System Prompt for Paraphrasing
Generate ONE paraphrase of the following sentence.

Specifically, we denote the input of the paraphrase model, including the concatenation of the system prompt x_{sys} and the original prompt x_{ori} that the model is instructed to rewrite, as $[x_{sys}, x_{ori}] = (x_1, \dots, x_n)$, where x_i is a token in the vocabulary V. Suppose the decoder of a pre-trained autoregressive language model P_{para} generates continuations of length m as the requested paraphrase, denoted as $x_{para} = (x_{n+1}, \dots, x_{n+m})$, based on the system prompt x_{sys} and original prompt x_{ori}. At decoding time, P_{para} iteratively decodes one token at a time by conditioning on the preceding context:

$$P_{para}(x_{para}|x_{sys}, x_{ori}) = \prod_{i=n+1}^{n+m} P_{para}(x_i|x_{<i}),$$

where $P_{para}(x_i|x_{<i})$ is the predicted next token probability.

2.3 Ensemble-based (Greedy) MONOPARA

To introduce perplexity constraint from the target model to the paraphrase model, here we introduce the ensemble-based greedy decoding for monotonic paraphrasing (as shown in the lower half of Fig. 2). Based on the intuition delivered in §2.1, MONOPARA decodes the paraphrase iteratively, and the next token is selected based on the combination of predicted probabilities from the two LMs:

$$\begin{aligned} x_{next} = \underset{x_i \in V}{argmax}(&\alpha \cdot \log P_{tar}(x_i|x_{gen}) \\ &+ (1-\alpha) \cdot \log P_{para}(x_i|x_{sys}, x_{ori}, x_{gen})), \end{aligned}$$

where x_{gen} denotes the sequence of generated tokens, and α determines the coefficient between the two models (Alg. 1).

By combining the prediction probabilities of both models, the ensemble can navigate the complex landscape of language generation more effectively and precisely. The paraphrase model P_{para}

Algorithm 1 Ensemble-based Decoding

Require:

x_{sys} ▷ System prompt for paraphrase
V ▷ Vocabulary
P_{tar}, P_{para} ▷ Target and paraphrase models
α ▷ Weighting factor

Ensure:

x_{gen} ▷ Generated text sequence

Initialize $x_{gen} \leftarrow \emptyset$
$x_{next} \leftarrow \underset{x_i \in V}{argmax}\ P_{para}(x_i | x_{sys} \oplus x_{ori})$ ▷ Generate first token with P_{para}
$x_{gen} \leftarrow x_{gen} \oplus x_{next}$
while not end of sequence **do**
  Compute weighted log probabilities for each $x_i \in V$:
  $score_{tar} \leftarrow \alpha \cdot \log P_{tar}(x_i | x_{gen})$
  $score_{para} \leftarrow (1-\alpha) \cdot \log P_{para}(x_i | x_{sys} \oplus x_{ori} \oplus x_{gen})$
  $x_{next} \leftarrow \underset{x_i \in V}{argmax}\ (score_{tar} + score_{para})$
  $x_{gen} \leftarrow x_{gen} \oplus x_{next}$
end while
return x_{gen}

keeps the rewritten prompt (or instruction) remains with the original intent, while the target model P_{tar} evaluates the paraphrase's likelihood or confidence, up-weighting any generation samples that lead towards low-perplexity outcomes. Based on the ensemble, the next token is selected by maximizing the weighted sum of logarithmic probabilities from both models, considering both the instruction and the generated text thus far.

As a coefficient, α balances the contribution of each model, ensuring that the final output satisfies both **semantic fidelity** to the prompt intent and **linguistic familiarity** to the target model. Higher α increases the weight of the logarithmic probability derived from the target model P_{tar} and cuts down on the semantic hints for P_{para}, which raises the risk of sacrificing the semantic fidelity of the generation. Conversely, a lower α shifts the emphasis towards P_{para}, placing greater value on ensuring that the generated text closely follows the instructional context and semantic intent from P_{para}. In this case, the decoding process lacks the constraint of perplexity from P_{tar} and sacrifices the linguistic familiarity of the generated paraphrase for the target model.

2.4 Search-based MonoPara

To further enhance the efficiency of decoding prompts of lower perplexity, we define the search-based decoding strategy for MonoPara (as shown in the upper half of Fig. 2). To take more steps of token generation into account and expand the search space, we leverage look-ahead decoding to search for the sequence that exhibits the lowest perplexity from an even broader candidate space. Following Lu et al. (2022a), each step of our search-based decoding consists of (i) expanding a set of candidate next-tokens, (ii) scoring each candidate, and (iii) selecting the k best candidates (Alg. 2 in Appx. §A):

$$X'_m = \{\mathrm{x}_{<m} \circ x_m | \mathrm{x}_{<m} \in X_{m-1}, x_m \in V\},$$
$$X_m = \underset{(x_{<m}, x_m) \in X'_m}{arg\ topk} \{f(\mathrm{x}_{<m}, x_m)\},$$

where x_m is a candidate predicted by paraphrase model, and $f(\cdot)$ is a scoring function that returns the perplexity of $\mathrm{x}_{<m} \circ x_m$ evaluated by the target model:

$$f(\mathrm{x}_{<m}, x_m) = \left(\prod_{i=1}^{m} \frac{1}{P_{tar}(x_i | x_{<i})} \right)^{\frac{1}{m}}. \quad (1)$$

Specifically, at each step m of the decoding process, the current set of sequences X_{m-1} is expanded by appending the most probable next-token x_m, predicted by the paraphrase model P_{para}, from vocabulary V to each sequence $\mathrm{x}_{<m}$. This results in a temporary expanded set of candidate sequences X'_m. Each candidate sequence in X'_m is then evaluated by the scoring function $f(\cdot)$ shown in Eq. 1, which calculates the perplexity of a sequence by the target model P_{tar}. Since a paraphrase of lower perplexity is preferred, the k sequences with the best scores (i.e., lowest perplexity) are selected to form the set X_m, narrowing down the choices to the k most promising sequences for continuation.

In comparison to greedy decoding, this look-ahead decoding mechanism evaluates a broader range of potential future impact from current choices (i.e., the selection of x_m), thereby allowing for a more informed and potentially more accurate selection of candidates.

3 Experiment

In this section, we demonstrate two distinct experimental settings for evaluation. In §3.1, we examine MonoPara's effectiveness on prompt refinement for eliciting better task performance. In §3.2, we evaluate MONOPARA's effect on enhancing model robustness and generalization under instruction perturbations.

3.1 Task I: Prompt Refinement

Task Description To inspect if monotonic paraphrasing can refine a prompt and elicit better performance on prompting an LLM as designed, we use SUPER-NATURALINSTRUCTION (SUP-NATINST for short, Wang et al., 2022). This benchmark consists of 1,616 diverse NLP tasks and their expert-written prompts for evaluating the zero-shot generalizability of LLMs on a variety of NLP tasks. In SUP-NATINST, each task is paired up with an instruction that consists of the task definition for mapping an input text to a task output and several examples for demonstrating the desired or undesired output. To make our evaluation more challenging, we only use the task definition as a prompt for the target model to perform the task. During test time, the target model is prompted with a concatenation of SUP-NATINST task description (or its paraphrase), test sample, and multiple choices of answers. Since SUP-NATINST provides only one description of definition for each task, we use GPT4 (Achiam et al., 2023) to generate 4 more SUP-NATINST-like task descriptions for each involved task, which results in 5 prompts for each task in total.

Datasets Following Gonen et al. (2023), we choose 7 classification tasks from Huggingface Datasets [3] to have a set of diverse tasks: (i) GLUE Cola (Warstadt et al., 2019) for grammatical acceptability discrimination; (ii) Newspop (Moniz and Torgo, 2018) for news classification; (iii) AG News (Zhang et al., 2015) for news classification; (iv) IMDB (Maas et al., 2011) for movie review sentiment analysis; (v) DBpedia (Lehmann et al., 2015) for topic classification; (vi) Emotion (Saravia et al., 2018) for tweet emotion classification; and (vii) Tweet Offensive (Barbieri et al., 2020) for offensive tweet discrimination. We sample 1,000 examples from each dataset for prompt evaluation. For these tasks, the prompt follows the input, and at the end of each prompt, we add the choices of classes following Gonen et al. (2023): "Choices: X, Y, Z. Answer: " as it helps with accuracy.

[3] https://huggingface.co/docs/datasets/index

Evaluation Metrics To inspect whether monotonic paraphrasing can refine a prompt into its low-perplexity counterpart and further enhance the generalization of LLM prompting, we use PPL (perplexity), Acc (accuracy), and BS (BERTScore, Zhang* et al., 2020) for evaluation. (1) Perplexity. Following Gonen et al. (2023), we define the perplexity of the prompt as the perplexity of the full prompt sequence, including the input itself and the choices of labels. To avoid noise when computing perplexity, the prompts are instantiated with 1,000 random examples of the dataset (we keep the same selected examples for performance evaluation), and the perplexity is averaged across all instantiated prompts. The correlation between the perplexity of a standalone prompt and the average perplexity over instantiated ones is illustrated in Appx. §E. (2) BERTScore. As a paraphrasing scheme, MONOPARA is also evaluated by its performance in semantic alignment with the source sentence based on pre-trained BERT (Devlin et al., 2019). (3) Accuracy. To compute accuracy, for each test sample, we obtain the model's predicted probability of all classes and choose the highest ranking class as the prediction of the model. The accuracy is calculated over the 1,000 samples for each task.

Models We study two auto-regressive models, Mistral-7B[4] (Jiang et al., 2023)and Starling-7B[5] (Zhu et al., 2023a), for their significant ability in instruction following and paraphrasing. In this paper, we keep the paraphrase model **the same** as the target model. Please refer to Appx. §B for the results on Llama-3-8B[6] (AI@Meta, 2024).

Baselines We compare our method with two baselines: SPELL (Gonen et al., 2023) and vanilla LLM-based paraphrasing (Para). Based on similar intuition that low perplexity and better performance exhibit strong correlation, SPELL (**S**electing **P**rompts by **E**stimating **L**M **L**ikelihood) ranks and selects the prompts with the lowest perplexity for a given task after creating a set of prompt candidates manually and expanding them to hundred-scale using automatic paraphrasing and back-translation. We use the prompt candidates

[4] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[5] https://huggingface.co/berkeley-nest/Starling-LM-7B-alpha

[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B

Methods	Metric	GLUE Cola	Newspop	AG News	IMDB	DBpedia	Emotion	Tweet Offensive
				Mistral 7B				
Ori.	Acc	61.76 ± 2.93	85.28 ± 3.45	67.68 ± 3.47	84.12 ± 4.89	70.82 ± 2.27	44.32 ± 1.57	**67.04** ± 0.95
	PPL	15.72 ± 4.60	12.60 ± 2.40	12.74 ± 1.07	16.41 ± 0.80	8.10 ± 0.22	16.48 ± 2.14	19.25 ± 3.43
SPELL	Acc	43.42 ± 1.16	81.08 ± 7.2	**73.88** ± 3.18	86.00 ± 0.81	71.02 ± 1.70	46.80 ± 1.12	65.06 ± 1.04
	PPL	8.24 ± 0.88	15.05 ± 4.15	15.99 ± 3.77	15.76 ± 1.42	25.36 ± 6.80	15.67 ± 5.21	14.31 ± 0.93
Para	Acc	39.48 ± 8.19	76.72 ± 16.54	71.60 ± 2.70	86.60 ± 4.86	70.22 ± 5.71	43.65 ± 3.41	65.04 ± 1.46
	PPL	78.01 ± 11.22	49.37 ± 7.81	28.37 ± 2.58	22.27 ± 0.98	28.34 ± 1.36	46.53 ± 5.30	53.65 ± 1.92
	BS	88.88 ± 0.74	87.47 ± 0.84	86.32 ± 2.08	90.23 ± 0.99	86.96 ± 0.93	90.08 ± 0.35	89.93 ± 0.67
Mono-E	Acc	**63.96** ± 3.88	**88.96** ± 3.04	72.20 ± 3.87	**87.58** ± 3.27	72.07 ± 3.54	**47.76** ± 3.72	67.00 ± 1.21
	PPL	14.89 ± 2.75	8.33 ± 1.30	10.07 ± 2.93	13.05 ± 1.86	8.85 ± 0.88	13.67 ± 1.87	14.11 ± 3.53
	BS	93.03 ± 2.15	92.85 ± 2.35	88.70 ± 2.35	89.35 ± 2.60	93.21 ± 1.16	92.74 ± 2.75	91.73 ± 1.32
Mono-S	Acc	62.31 ± 6.17	86.34 ± 5.16	70.80 ± 3.44	83.48 ± 2.87	**72.46** ± 3.00	44.26 ± 3.05	65.66 ± 2.43
	PPL	15.85 ± 4.88	11.95 ± 2.09	9.89 ± 1.31	16.38 ± 1.63	8.63 ± 0.41	15.03 ± 1.65	18.35 ± 4.07
	BS	92.47 ± 2.00	93.78 ± 2.68	91.57 ± 2.23	89.66 ± 3.25	95.87 ± 1.20	93.42 ± 1.02	91.62 ± 2.48
				Starling 7B				
Ori.	Acc	55.18 ± 12.43	93.78 ± 1.18	**88.44** ± 1.18	96.46 ± 0.17	**94.06** ± 0.37	55.48 ± 0.91	64.62 ± 0.18
	PPL	16.38 ± 4.18	12.77 ± 2.59	12.10 ± 0.91	15.56 ± 0.57	8.11 ± 0.22	17.90 ± 2.82	20.41 ± 3.90
SPELL	Acc	46.46 ± 6.70	89.84 ± 0.91	83.72 ± 0.26	92.84 ± 2.7	92.02 ± 0.64	54.82 ± 1.81	64.20 ± 0.17
	PPL	44.96 ± 18.23	54.60 ± 18.72	16.81 ± 0.91	56.34 ± 15.67	28.66 ± 9.47	32.79 ± 8.53	33.56 ± 3.83
Para	Acc	40.38 ± 15.49	82.67 ± 19.63	86.39 ± 2.92	**96.66** ± 0.34	93.10 ± 1.79	51.10 ± 10.18	64.44 ± 0.08
	PPL	70.50 ± 16.38	54.41 ± 8.80	30.67 ± 2.81	21.78 ± 1.37	26.56 ± 1.39	58.87 ± 4.44	11.71 ± 0.86
	BS	90.35 ± 0.67	88.35 ± 0.51	88.46 ± 0.98	90.04 ± 1.04	87.06 ± 0.79	90.38 ± 0.45	79.93 ± 11.52
Mono-E	Acc	**69.62** ± 4.12	**95.15** ± 0.59	85.40 ± 4.22	96.22 ± 0.47	94.05 ± 0.42	**56.10** ± 1.08	64.45 ± 0.09
	PPL	8.56 ± 1.00	10.12 ± 1.02	8.52 ± 0.76	12.60 ± 1.57	9.58 ± 0.86	9.23 ± 0.97	11.71 ± 0.86
	BS	93.00 ± 0.85	94.19 ± 0.94	91.06 ± 1.51	92.85 ± 1.39	93.23 ± 0.50	93.28 ± 0.26	93.69 ± 1.52
Mono-S	Acc	50.96 ± 11.52	88.20 ± 3.74	83.61± 6.30	92.96 ± 4.78	92.82 ± 2.17	54.90 ± 2.32	**66.80** ± 1.22
	PPL	14.05 ± 3.76	12.55 ± 2.07	10.61 ± 1.06	16.44 ± 0.92	8.39 ± 0.53	16.28 ± 2.18	20.17 ± 3.17
	BS	92.69 ± 0.99	95.83 ± 1.66	93.92 ± 2.23	93.01 ± 1.25	96.99 ± 0.70	94.78 ± 1.25	94.45 ± 1.43

Table 1: Results of Task I (Prompt Refinement). Ori. refers to the original prompts. Para refers to paraphrasing the original prompts without perplexity constraint. Mono-E refers to ensemble-based MONOPARA. Mono-S refers to search-based MONOPARA. The best accuracy for each task is in **bold**, while the lowest perplexity and the highest BERTScore for each task is in blue and orange respectively. The coefficient α is fixed as 0.5 for all the results.

provided by Gonen et al. (2023) and rank them by the perplexity w.r.t. the target model. Para paraphrases source prompts with greedy decoding (top-1 sampling) by directly prompting the target LLM with the instruction introduced in §2.2. Besides, we also implement reranking based on the prompt candidates generated by MONOPARA for a fair comparison with SPELL. The implementation details and results are discussed in Appx. §D.

Results As shown in Tab. 1, the two variants of MONOPARA successfully refine prompts to those with perplexities lower than the original ones, thereby enhancing model performance on 10 out of 14 scenarios (2 models, each on 7 datasets). Specifically, Mono-E (ensembled-based MONOPARA) improves the accuracy of Mistral-7B by an average of 2.64% and that of Starling-7B by an average of 1.85% on seven different datasets. In contrast, directly paraphrasing the prompts without perplexity constraint (Para) results in prompts with higher perplexities than the original ones in 13 out of 14 scenarios, most of which lead to worse task performance. This highlights the importance of incorporating perplexity constraints when paraphrasing prompts. The correlation between task accuracy and prompt perplexity is analyzed in Appx. §C.

Besides, our method outperforms SPELL with consistently lower perplexity and superior task performance with the refined prompt. This superiority stems from the fact that, although SPELL explores a vast search space with hundred-scale prompt candidates, the candidates are searched in an ad hoc manner without an optimization goal, leading to inefficiencies in prompt rewriting or paraphrasing. In contrast, our method incorporates the perplexity constraint that leads to optimized target-model perplexity during one single refined prompt generation process, leading to both enhanced task performance and refinement efficiency.

Moreover, MonoPara achieves superior performance in terms of both average prompt perplexity and BERTScore, indicating the synergistic role played by both the target model and the paraphrase model in prompt refinement. During decoding, the target model prioritizes achieving low perplexity, while the paraphrase model preserves the original prompt intent, resulting in a high BERTScore. We provide a detailed case study on BERTScore of Para and MonoPara in Appx. §G to explain why MonoPara outperforms Para on BERTScore. Specifically, Mono-E outperforms Mono-S in some cases since Mono-S involves the additional hyperparameter of beam size which we did not tune. Additionally, having additional token options at each decoding step may introduce more randomness, potentially affecting the overall performance. That being said, Mono-S mostly exhibits a higher BERTScore than the simpler method Mono-E, indicating better quality of paraphrasing.

3.2 Task II: Robustness Evaluation on Instruction Perturbation

We further evaluate MonoPara's effectiveness in enhancing model robustness against various instruction perturbations at character, word, sentence, and semantic levels using PromptBench (Zhu et al., 2023b).

Task Description PromptBench introduces perturbation to task instructions of a diverse set of tasks, such as sentiment analysis, grammar correctness, duplicate sentence detection, and natural language inference. It includes character-level perturbations using DeepWordBug (Gao et al., 2018) to introduce typos, word-level using TextFooler (Jin et al., 2020) to replace words with contextually similar words, and semantic-level by using prompts following the linguistic behavior of different languages. For semantic-level perturbations, the adversary constructs prompts using various languages, such as Chinese, French, Arabic, Spanish, Japanese, and Korean, and then translates these prompts into English. By exploiting the nuances and idiosyncrasies of different languages during translation, it can introduce subtle ambiguities, grammatical errors, or inconsistencies in the input prompt. The perturbed prompt itself is still in English.

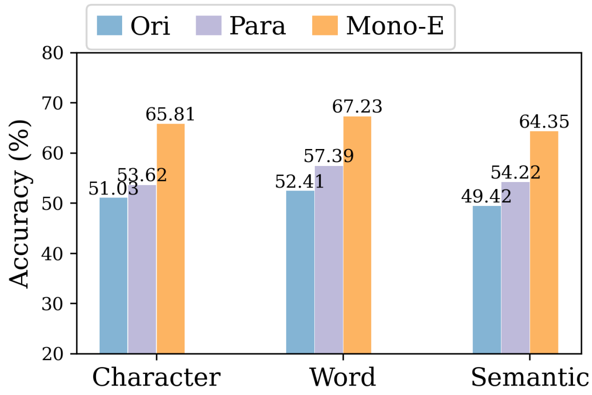

Figure 3: Model's average accuracy across 4 GLUE datasets, with each dataset having six instructions with perturbation added at character, word, and semantic levels. Mono-E has consistent improvement in accuracy across all types of perturbation compared to vanilla paraphrasing.

Datasets We select 4 GLUE tasks (Wang et al., 2018) from Huggingface Datasets for evaluation: (i) Cola, which is also used for Task I; (ii) SST-2 (Socher et al., 2013) for two-way sentiment classification; (iii) MNLI (Wang et al., 2018) for multi-genre inference identification; and (iv) MRPC (Dolan and Brockett, 2005) for paraphrase identification. We adopt the same instruction format as that of Task I.

Model For this task, we use Alpaca[7] (Taori et al., 2023), a model instruction-tuned from the LLaMA model (Touvron et al., 2023a) on the 52k self-instruct dataset. The paraphrase model is also the same as the target model.

Results As shown in Fig. 3, we compare the performance of original prompts, prompts paraphrased without perplexity constraint, and our method Mono-E. Across all kinds of instruction perturbations, our method achieves consistent accuracy improvement by up to 10.13%. This notable increase in performance demonstrates that MonoPara not only refines prompts for improved task performance but also enhances model robustness against various instruction perturbations. We also show the model performance with unperturbed prompts and compare the degradation from perturbation in Appx. §F to show the robustness of Mono-E.

[7] `https://huggingface.co/tloen/alpaca-lora-7b`. We use a different language model for this task to demonstrate the robustness and versatility of our proposed method across various models.

		Content	BertScore	PPL
Ori Instruction		Read the provided excerpt and choose between ‘positive’ and ‘negative’ to describe its sentiment Iyk1lJt8yw:	-	78.12
Para		Determine the sentiment of the given excerpt as either positive or negative	81.26	44.29
Mono	$\alpha = 0.2$	Determine the sentiment of the given text by selecting either ‘positive’ or ‘negative’.	86.18	24.90
	$\alpha = 0.5$	Determine the sentiment of the given text by classifying it as positive or negative.	80.22	16.27
	$\alpha = 0.7$	Determine the remainder when 101 is divided by 10.\n A: 1	50.22	4.53

Table 2: Effect of coefficient α on Mono-E with Mistral-7B model.

3.3 Effect of Coefficient

We further analyze the effect of coefficient α between the target model and the paraphrase model in MONOPARA. Tab. 2 presents a case study of α on the ensemble-based decoding variant (Mono-E) with a prompt under character-level perturbation. Obviously, a larger value of α amplifies the influence of the target model, resulting in a stronger perplexity constraint and monotonically lower perplexity. Consequently, the impact of the paraphrase model is diminished, leading to a lower BERTScore. By adjusting α, the user can find a balanced point to achieve better prompts that strike a balance between semantic fidelity to the original prompt intent and linguistic familiarity to the target model. Detailed results and analysis are presented in Appx. §H and Tab. 9.

4 Related Work

Paraphrasing. Paraphrase generation has been a longstanding task in NLP. Since the era of deep learning, studies have adopted the universal paradigm of training sequence-to-sequence generation models based on abundant learning resources (Nema et al., 2017; Gupta et al., 2018; Li et al., 2018). While more recent works have leveraged more robust pre-trained language models that captured rich supervision signals from various sequence-to-sequence generation tasks (Raffel et al., 2020; Lewis et al., 2020) and achieving even more robust performance of paraphrasing with relatively limited end-task supervision (Sun et al., 2021; Meng et al., 2021; Bui et al., 2021). In addition to these advancements, a series of works have added auxiliary supervision signals such as syntactical guidance, lexical regularization, and cross-lingual supervision, to further improve the performance and controllability of paraphrasing in aspects such as syntactic structures, stylistic specification, or textual simplicity (Iyyer et al., 2018; Li et al., 2019; Chen et al., 2019; Kumar et al., 2020; Goyal and Durrett, 2020; Hosking and Lapata, 2021; Chowdhury et al., 2022).

Different from existing paraphrasing techniques, this work conducts constrained decoding methods that seek to monotonically decrease the perplexity in the process of paraphrasing. This innovative approach serves as a general approach to rewrite and refine prompts or task instructions towards their more familiar counterparts for LMs, seeking to enhance the generalization of LM prompting without the need for any training effort.

Prompt Refinement. Prompt refinement has been actively explored in recent years with the aim of selecting, retrieving, or even generating prompts that lead to improved zero-shot performance of LM prompting. When an end-task objective is known, prior works on prompt refinement approaches often set the end-task performance as the objective, and leverage approaches such as gradient-based search, similar to continuous prompts but with projections onto a discrete vocabulary (Shin et al., 2020). Other works have explored edit-based enumeration (Prasad et al., 2023), reinforcement learning (Deng et al., 2022), and large language model continuation and filtering (Zhou et al., 2022). As for general-purpose LLMs, a dedicated prompt refinement approach may not assume the pre-existence of any specific end tasks. Hence, more recent works Iter et al. (2023); Gonen et al. (2023); Wang et al. (2023); Lee et al. (2023) explored the heuristic criterion based on the familiarity of the language model with the language contained in the prompt, as measured by its perplexity. Lower perplexity prompts are preferred as they are expected to perform better across a wide range of tasks.

While the last line of research has relieved the generic principle for improving zero-shot LLM prompting generalizability, these studies are largely limited to selecting among pre-existing prompts or multiple beam search samples. Our proposed monotonic paraphrasing approach, however, can proactively rewrite a prompt or task instruction into counterparts that better satisfy the above heuristic criterion, without the need of any pre-existing candidate prompts or any training effort.

Ensemble and Search-based Decoding. The proposed two decoding schemes of MONOPARA are connected to both ensemble decoding and search-based decoding approaches in recent studies. Ensemble decoding has been previously proposed for purposes including debiasing (Li et al., 2023), controllable generation (Meng et al., 2022; Huang et al., 2023), and cross-modality data processing (Liu et al., 2023). Search-based decoding, on the other hand, is proposed to efficiently extend the search space of generation samples for satisfying specific constraints or criteria (Lu et al., 2022a; Wuebker et al., 2012; Lee and Berg-Kirkpatrick, 2022; Xu et al., 2023). Representative approaches include look-ahead (Lu et al., 2022a) and look-back (Xu et al., 2023) ones that search within the beam space towards different directions.

The explored decoding schemes in MONOPARA are inspired by both lines of studies on decoding approaches. Specifically, the first line is related to our design choice for composing the probabilities of both the paraphrasing and the target LMs in greedy generation. And our search-based decoding scheme is specifically inspired by the look-ahead decoding strategy in the second line of studies.

5 Conclusion

In this paper, we propose monotonic paraphrasing (MONOPARA) for automatically and iteratively generating low-perplexity prompts for given LLMs, which can lead to better task performance. Following the high-level idea of paraphrasing prompts with perplexity constraints from the target LLM, we design ensemble-based and search-based decoding strategies for efficient prompt refinement. Experiments on prompt variations of unseen tasks and instructions demonstrate the effectiveness of MONOPARA in reducing prompt perplexity while enhancing task performance and model robustness. Future work may apply MONOPARA on other scenarios, such as searching stealthy prompts for red-teaming. Replacing perplexity with other criteria, such as complexity (Li et al., 2024), for different purposes of constrained generation is also a meaningful research direction.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. Qin Liu was supported by a departmental fellowship. Fei Wang was supported by the Amazon ML Fellowship. Tianyi Yan was supported by the CURVE Fellowship. Muhao Chen was supported by the DARPA FoundSci Grant HR00112490370, the NSF of the United States Grant ITE 2333736, and an Amazon Research Award.

Limitations

The current investigation of MONOPARA has the following limitations. First, using the target model as the paraphrase model is a natural technical choice. However, examining other paraphrasing models, especially those customized for paraphrasing, may lead to better performance. Second, our proposed method can not be applied to black-box LLMs such as ChatGPT[8], PaLM (Chowdhery et al., 2023), Claude[9], etc. This limitation arises from the necessity of access to the decoding phase, which in turn comes with far less computational cost due to zero training, and does not require the pre-existence of multiple prompt or instruction candidates. Third, while we conduct experiments on a wide range of tasks, their original prompts and responses are relatively short. This actually limits the operational space of MONOPARA. Experiments on tasks with longer inputs and outputs may provide additional evidence of MONOPARA's effectiveness.

References

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

AI@Meta. 2024. Llama 3 model card.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

[8]https://chat.openai.com

[9]https://www.anthropic.com/claude

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tien-Cuong Bui, Van-Duc Le, Hai-Thien To, and Sang Kyun Cha. 2021. Generative pre-training for paraphrase generation by representing and predicting spans in exemplars. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 83–90. IEEE.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

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

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

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10535–10544.

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.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. Association for Computational Linguistics.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. ArXiv:1801.04354 [cs].

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.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Tom Hosking and Mirella Lapata. 2021. Factorising meaning and form for intent-preserving paraphrasing. 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 1405–1418, Online. Association for Computational Linguistics.

Tenghao Huang, Ehsan Qasemi, Bangzheng Li, He Wang, Faeze Brahman, Muhao Chen, and Snigdha Chaturvedi. 2023. Affective and dynamic beam search for story generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11792–11806, Singapore. Association for Computational Linguistics.

Dan Iter, Reid Pryzant, Ruochen Xu, Shuohang Wang, Yang Liu, Yichong Xu, and Chenguang Zhu. 2023. In-context demonstration selection with cross entropy difference. *arXiv preprint arXiv:2305.14726*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

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

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. ArXiv:1907.11932 [cs].

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Aswanth Kumar, Ratish Puduppully, Raj Dabre, and Anoop Kunchukuttan. 2023. CTQScorer: Combining multiple features for in-context example selection for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7736–7752, Singapore. Association for Computational Linguistics.

Ivan Lee and Taylor Berg-Kirkpatrick. 2022. HeLo: Learning-free lookahead decoding for conversation infilling. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4996–5008, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yoonsang Lee, Pranav Atreya, Xi Ye, and Eunsol Choi. 2023. Crafting in-context examples according to lms' parametric knowledge. *arXiv preprint arXiv:2311.09579*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bangzheng Li, Ben Zhou, Xingyu Fu, Fei Wang, Dan Roth, and Muhao Chen. 2024. Famicom: Further demystifying prompts for language models with task-agnostic performance estimation. *arXiv preprint arXiv:2406.11243*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. 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.

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.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414, Florence, Italy. Association for Computational Linguistics.

Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. 2023. Prismer: A vision-language model with multi-task experts. *Transactions on Machine Learning Research*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022a. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2022b. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022c. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

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

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. 2022. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35:28125–28139.

Yuxian Meng, Xiang Ao, Qing He, Xiaofei Sun, Qinghong Han, Fei Wu, Chun Fan, and Jiwei Li. 2021. ConRPG: Paraphrase generation using contexts as regularizer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2551–2562, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nuno Moniz and Luís Torgo. 2018. Multi-source social feedback of online news feeds. *arXiv preprint arXiv:1801.07055*.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

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.

KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *arXiv preprint arXiv:2302.12822*.

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.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zezhong Wang, Luyao Ye, Hongru Wang, Wai-Chung Kwan, David Ho, and Kam-Fai Wong. 2023. Read-Prompt: A readable prompting method for reliable knowledge probing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7468–7479, Singapore. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Joern Wuebker, Hermann Ney, and Richard Zens. 2012. Fast and scalable decoding with language model look-ahead for phrase-based statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 28–32, Jeju Island, Korea. Association for Computational Linguistics.

Nan Xu, Chunting Zhou, Asli Celikyilmaz, and Xuezhe Ma. 2023. Look-back decoding for open-ended text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1039–1050, Singapore. Association for Computational Linguistics.

Tianyi Yan, Fei Wang, James Y. Huang, Wenxuan Zhou, Fan Yin, Aram Galstyan, Wenpeng Yin, and Muhao Chen. 2024. Contrastive instruction tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10288–10302, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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 Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023a. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023b. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

Appendices

A Algorithm for Search-based MonoPara

Algorithm 2 Search-based Decoding

Require:

V ▷ Vocabulary

P_{tar}, P_{para} ▷ Target and paraphrase models

k ▷ Number of candidate sequences to retain

Ensure:

x_{best} ▷ Best generated sequence

$X_0 \leftarrow \{\emptyset\}$ ▷ Initialize with empty sequence

while not end **do**

  $m \leftarrow$ current decoding step

  $X'_m \leftarrow \emptyset$

  for each $\mathrm{x}_{<m} \in X_{m-1}$ **do**

    for each $x_m \in V$ **do**

      Append $x_{<m} \circ x_m$ to X'_m

    end for

  end for

  Use P_{para} for k candidates for each $\mathrm{x}_{<m}$

  $X_m \leftarrow \underset{(x_{<m}, x_m) \in X'_m}{arg\ topk} \{f(\mathrm{x}_{<m}, x_m)\}$ ▷ Select k candidates with lowest perplexity

  $end \leftarrow$ If any sequence ends with EOS

end while

$x_{best} \leftarrow$ select the sequence from X_m with the lowest perplexity by P_{tar}

return x_{best}

B Results on Llama-3-8B

Results on Llama-3-8B are illustrated in Tab. 3.

C Correlation Between Accuracy and Perplexity

To illustrate the interplay between task accuracy and prompt perplexity, we calculate the Pearson and Spearman correlation for each task based on the results in Tab. 1. The results are shown in Tab. 4. Most tasks show a strong negative correlation between perplexity and accuracy on both models except AG News, which is also discovered by Gonen et al. (2023) that for some of the tasks, there is not a negative correlation between perplexity and accuracy. The paraphrasing may be more helpful for cases where the nuances of expression don't matter so much (or there's enough room for error), but in domains where precision of language is important, perhaps perplexity-based paraphrasing might not be as helpful. AG News is a four-class news classification task with labels of "World", "Business", "Sci/Tech", and "Sports", which are not as easy as other binary classification tasks. As a result, the precision of the instruction for AG News is important. Additionally, we can see that the accuracy of AG News is not affected much under different methods, indicating that the accuracy is not so dependent on the perplexity of the prompt as other tasks. This indicates that MonoPara works best for tasks with high distinction among labels.

D Reranking for MonoPara

D.1 Implementation Setting

We implemented reranking on top of the linear combination of P_{tar} and P_{para} under two settings: (1) the coefficient α equals 0.5, which is kept consistent with the results presented in Tab. 1 for direct comparison; (2) the coefficient is randomly selected from $[0.1, 0.2, \ldots, 0.6]$ for a larger search space. In both cases, we allow sampling from the top 3 tokens at every decoding step for randomness.

For each task, we use the 5 prompts as the original prompts (the same setting as Tab. 1) and randomly generate 100 paraphrases with the linear combination of P_{tar} and P_{para} for every single original prompt. For the generated 100 paraphrases, we use the same scheme as is described in §3 to calculate the perplexity: each paraphrase is instantiated with $1,000$ random test samples, and the perplexity is averaged across all the $1,000$ instantiated prompts. For budget k, we select the first k paraphrases from the 100 candidates and choose the prompt with the lowest average instantiated perplexity to be tested on the task.

D.2 Analysis

For the reranking setting (1), with additional randomness when sampling paraphrases from the linear combination of P_{tar} and P_{para}, a large enough budget (i.e., 100 paraphrases) can consistently catch paraphrased prompts with lower perplexity which achieve better task performance (Tab. 5). However, with a small budget of 10 or even 50, reranking can hardly surpass Mono-E in either perplexity or task performance. The reason is that with randomness allowed, the paraphrase is no longer decoded greedily at each token, which undermines the constraint on the perplexity and leads to a large variation of the perplexity of the resulting paraphrase. This further affects the performance of

Methods	Metric	GLUE Cola	Newspop	AG News	Tweet Offensive
Ori.	Acc	54.82 ± 1.73	73.82 ± 4.57	60.37 ± 7.59	63.25 ± 0.95
	PPL	9.42 ± 5.08	17.72 ± 6.11	13.64 ± 4.97	23.64 ± 4.61
Para	Acc	43.69 ± 9.02	68.45 ± 13.24	64.85 ± 3.73	64.22 ± 1.48
	PPL	32.07 ± 8.43	29.41 ± 8.17	23.77 ± 4.32	46.14 ± 9.24
SPELL	Acc	46.91 ± 2.33	71.67 ± 5.30	66.93 ± 3.70	64.73 ± 0.76
	PPL	7.98 ± 1.58	12.47 ± 3.76	11.29 ± 2.79	8.79 ± 1.93
Mono-E	Acc	58.35 ± 2.61	77.85 ± 4.01	65.45 ± 2.11	64.90 ± 0.59
	PPL	6.64 ± 0.95	9.17 ± 2.59	8.29 ± 1.47	6.40 ± 3.61

Table 3: Performance of Llama-3-8B.

Model	**Metric**	**GLUE Cola**	**Newspop**	**AG News**	**IMDB**	**DBpedia**	**Emotion**	**Tweet Offensive**
Mistral	Pearson	-0.62	-0.88	0.19	-0.04	-0.70	-0.57	-0.48
	Spearman	-0.30	-0.99	0.30	-0.30	-0.39	-0.70	-0.30
Starling	Pearson	-0.83	-0.69	0.16	-0.60	-0.70	-0.96	-0.05
	Spearman	-0.90	-0.49	0.3	-0.3	-0.70	-0.89	-0.15

Table 4: Pearson and Spearman correlation between task accuracy and prompt perplexity w.r.t. the target model. Results are calculated based on Tab. 1.

these paraphrased prompts, which in turn proves the correlation between perplexity and task accuracy. Generally, we can expect vanilla Mono-E and reranking to break even with a large budget of around 50 paraphrases.

The reranking setting (2) is not as effective as setting (1), where even a budget of 100 could not surpass the vanilla Mono-E. The reason is that although a large search space is allowed with randomness on the combination coefficient α, the constraint on perplexity while decoding is undermined even further. With more randomness for paraphrase sampling, the results of setting (2) also show a higher standard deviation than setting (1).

E Correlation Between Standalone and Instantiated Prompts

To bridge the intuition gap, we investigate the correlation between the standalone prompt and its instantiated prompts. We instantiate each task-specific prompt with $1,000$ input instances and the perplexity used for correlation evaluation is averaged over 5 prompts for each task. As shown in Tab. 6, the Pearson correlation is high (close to 1.0) in most cases for different datasets despite slight noise for Emotion under greedy paraphrase of prompts. We can conclude that there is a high correlation between instantiated and standalone prompts. Thus, it is valid to use the perplexity of the standalone prompt optimized by P_{tar} to serve as the constraint.

F Unperturbed Accuracy for Task II

Tab. 7 shows the task performance of three perturbation methods (Alpaca's average accuracy across 4 GLUE datasets) and performance degradation compared with the unperturbed prompt (as shown in the brackets). We can witness a narrow performance gap between perturbed and unperturbed original prompts for Mono-E, which is more robust than directly using the given prompt (Ori) or the greedily paraphrased one (Para). The reason is that Mono-E largely mitigates the effect of adversarial perturbations on the prompts and successfully rewrites the given prompts (whether perturbed or unperturbed) into higher-quality versions.

G BERTScore of Para and MonoPara

We provide a case study on the paraphrase results and the according BERTScore in Tab. 8. We could see that both Para and Mono-E largely preserve the semantic meaning of the original prompt. Further, Para tends to use new vocabulary for diversity, while Mono-E prefers words that are more familiar to the target model, which is often the same as

Method	Metric	GLUE Cola	Newspop	AG News	IMDB	DBpedia	Emotion	Tweet Offensive
Ori.	Acc	61.76 ± 2.93	85.28 ± 3.45	67.68 ± 3.47	84.12 ± 4.89	70.82 ± 2.27	44.32 ± 1.57	67.04 ± 0.95
	PPL	15.72 ± 4.60	12.60 ± 2.40	12.74 ± 1.07	16.41 ± 0.80	8.10 ± 0.22	16.48 ± 2.14	19.25 ± 3.43
Para	Acc	39.48 ± 8.19	76.72 ± 16.54	71.60 ± 2.70	86.60 ± 4.86	70.22 ± 5.71	43.65 ± 3.41	65.04 ± 1.46
	PPL	78.01 ± 11.22	49.37 ± 7.81	28.37 ± 2.58	22.27 ± 0.98	28.34 ± 1.36	46.53 ± 5.30	53.65 ± 1.92
Mono-E	Acc	63.96 ± 3.88	*88.96 ± 3.04*	*72.20 ± 3.87*	*87.58 ± 3.27*	*72.07 ± 3.54*	**47.76 ± 3.72**	67.00 ± 1.21
	PPL	14.89 ± 2.75	*8.33 ± 1.30*	*10.07 ± 2.93*	*13.05 ± 1.86*	8.85 ± 0.88	*13.67 ± 1.87*	*14.11 ± 3.53*
Rerank (1)								
Budget = 10	Acc	62.20 ± 3.47	86.14 ± 3.14	69.31 ± 4.29	85.02 ± 4.13	69.88 ± 4.33	47.36 ± 3.28	66.12 ± 2.15
	PPL	14.27 ± 2.69	11.46 ± 1.99	11.92 ± 3.97	15.95 ± 2.83	13.15 ± 1.24	15.06 ± 2.71	15.63 ± 3.57
Budget = 50	Acc	63.84 ± 2.46	85.86 ± 5.36	68.28 ± 5.16	86.52 ± 5.86	69.64 ± 3.53	47.60 ± 5.25	67.74 ± 1.45
	PPL	13.54 ± 3.16	10.53 ± 2.85	10.83 ± 3.24	13.44 ± 2.22	10.31 ± 1.85	14.14 ± 3.46	13.94 ± 3.86
Budget = 100	Acc	**64.58 ± 3.50**	**89.82 ± 2.04**	**72.33 ± 4.17**	**87.61 ± 4.93**	**72.52 ± 3.16**	*47.70 ± 3.56*	**68.5 ± 1.84**
	PPL	**10.22 ± 2.32**	**7.53 ± 1.82**	**9.52 ± 4.20**	**12.88 ± 3.01**	**8.16 ± 1.67**	**13.05 ± 1.34**	**11.56 ± 4.44**
Rerank (2)								
Budget = 10	Acc	62.35 ± 4.77	86.96 ± 4.61	70.02 ± 5.27	84.86 ± 4.71	69.31 ± 3.94	46.10 ± 7.29	66.63 ± 4.53
	PPL	15.63 ± 3.85	12.27 ± 3.99	13.47 ± 3.63	16.43 ± 2.17	15.37 ± 2.14	16.64 ± 3.88	18.87 ± 4.48
Budget = 50	Acc	62.08 ± 3.97	86.50 ± 5.10	71.12 ± 6.97	86.03 ± 5.24	70.13 ± 4.23	46.90 ± 5.83	65.15 ± 5.03
	PPL	14.42 ± 4.16	10.24 ± 4.36	11.69 ± 2.96	14.33 ± 3.55	13.62 ± 2.77	15.21 ± 3.38	18.53 ± 4.54
Budget = 100	Acc	*64.35 ± 4.91*	87.82 ± 4.48	71.72 ± 6.07	86.64 ± 5.16	71.98 ± 3.85	47.18 ± 4.63	*67.80 ± 8.95*
	PPL	*11.67 ± 4.37*	8.93 ± 3.15	10.61 ± 4.89	13.47 ± 3.84	*8.36 ± 3.57*	14.41 ± 2.95	16.65 ± 3.53

Table 5: Reranking for ensemble-based MONOPARA. The best results of each task are **bolded** and the second-best are *italicized*.

	Newspop	IMDB	DBpedia	Emotion	Tweet
Ori	0.949	0.952	0.895	0.839	0.974
Para	0.813	0.951	0.835	0.645	0.871
Mono-E	0.956	0.895	0.927	0.931	0.968

Table 6: Pearson correlation between the perplexity of standalone prompt and its instantiated prompts. Pearson correlation is high (close to 1.0) in most cases, indicating a high correlation.

	Ori	Para	Mono-E
Un.	56.67	58.31	65.94
Cha.	51.03 (-9.95%)	53.62 (-8.04%)	65.81 (-0.20%)
Word	52.41 (-7.51%)	57.39 (-1.58%)	67.23 (+1.96%)
Sem.	49.42 (-12.79%)	54.22 (-7.01%)	64.35 (-2.4%)

Table 7: Performance degradation of Alpaca model compared with the unperturbed prompt. *Un.*, *Cha.*, and *Sem.* are short for unperturbed, character, and semantic, respectively, referring to different prompt perturbation methods.

in the original prompt. This might be the reason why Para results in lower BertScore compared to Mono-E and Mono-S.

H Ablation Study on Alpha Value

We provide results with various alpha values in Tab. 9, which illustrates the impact of alpha values on model accuracy and paraphrase perplexity across various datasets on the Mistral-7B model. With the increase of α value, the perplexity (PPL) of the paraphrase monotonically goes down due to the design and motivation of our method. Given the low perplexity, we compare the task accuracy of various alpha values. For α = 0.4 or 0.6, the model shows outstanding performance on some of the datasets. But increasing alpha to 0.7 results in a nonsensical prompt though the perplexity is super low. From the results we can also see that the performance just corrupted with this nonsensical prompt. The setting of α should seek a balance between lower perplexity and the preservation of the original semantics. As a result, alpha = 0.5 (Mono-E) offers a balanced and reliable performance, making it a suitable default choice.

	Content	BertScore
Ori Prompt	In this task, you are presented with a short article. Your objective is to classify the article according to its category using the following labels: World, Sports, Business, Sci/Tech. Tag the text as "World" if it involves information about global events or issues. Mark it as "Sports" if it deals with sports-related content. Label it "Business" if it pertains to business affairs, markets, or economics. Assign "Sci/Tech" if it covers scientific or technological developments.	–
Para	In this **assignment**, you are given a brief article. Your goal is to **categorize** the article based on its topic using these tags: World, Sports, Business, Sci/Tech. Identify it as "World" if it discusses **international** events or matters. Designate it as "Sports" if it focuses on athletic content. Categorize it as "Business" if it concerns commercial activities, markets, or the economy. Label it "Sci/Tech" if it reports on advancements in science or technology.	86.02
Mono-E	In your **task** you will have an input which consists of a single article. You have to sort this article according to its category using a list which consists of 4 categories: "World", "Sci/tech", "Business", or "Sports". You should mark it with the appropriate label based on the content. The article should be labeled with the "Sports" category, when its topic deals with sports. When the article is about business, you need to mark it as a 'business' label. If you are talking about science and tech developments then label the content with "Sci/Tech", if it is a **world-related** issue or event then label it with "World".	89.63

Table 8: Case study on the paraphrase results and BERTScore. Para tends to use new vocabulary for diversity, while Mono-E prefers words that are more familiar to the target model, which is often the same as in the original prompt.

Method	**Metric**	**GLUE Cola**	**Newspop**	**AG News**	**IMDB**	**DBpedia**	**Emotion**	**Tweet**
Ori.	Acc	61.76 ± 2.93	85.28 ± 3.45	67.68 ± 3.47	84.12 ± 4.89	70.82 ± 2.27	44.32 ± 1.57	67.04 ± 0.95
	PPL	15.72 ± 4.60	12.60 ± 2.40	12.74 ± 1.07	16.41 ± 0.80	8.10 ± 0.22	16.48 ± 2.14	19.25 ± 3.43
Para	Acc	39.48 ± 8.19	76.72 ± 16.54	71.60 ± 2.70	86.60 ± 4.86	70.22 ± 5.71	43.65 ± 3.41	65.04 ± 1.46
	PPL	78.01 ± 11.22	49.37 ± 7.81	28.37 ± 2.58	22.27 ± 0.98	28.34 ± 1.36	46.53 ± 5.30	53.65 ± 1.92
Mono-E	Acc	*63.96 ± 3.88*	88.96 ± 3.04	*72.20 ± 3.87*	87.58 ± 3.27	*72.07 ± 3.54*	**47.76 ± 3.72**	67.00 ± 1.21
	PPL	14.89 ± 2.75	8.33 ± 1.30	10.07 ± 2.93	13.05 ± 1.86	8.85 ± 0.88	13.67 ± 1.87	14.11 ± 3.53
α = 0.1	Acc	60.80 ± 4.28	84.98 ± 3.41	69.09 ± 4.01	85.52 ± 1.72	70.42 ± 3.66	44.80 ± 2.31	65.56 ± 5.07
	PPL	15.61 ± 2.51	12.32 ± 1.01	12.03 ± 2.96	15.70 ± 1.85	9.61 ± 0.61	16.16 ± 1.11	18.49 ± 2.84
α = 0.2	Acc	61.80 ± 5.63	85.32 ± 3.56	71.49 ± 4.32	85.08 ± 4.97	70.54 ± 5.23	44.04 ± 2.44	66.80 ± 5.19
	PPL	15.22 ± 2.06	11.56 ± 1.27	11.68 ± 3.25	15.30 ± 1.55	9.23 ± 0.79	15.71 ± 0.20	16.92 ± 2.68
α = 0.3	Acc	61.54 ± 5.85	86.72 ± 4.94	**72.4 ± 2.77**	86.68 ± 4.21	71.90 ± 3.91	44.96 ± 2.67	*67.56 ± 3.59*
	PPL	15.18 ± 2.84	10.21 ± 1.22	11.06 ± 2.74	14.42 ± 1.05	8.95 ± 0.75	15.21 ± 1.04	15.87 ± 3.61
α = 0.4	Acc	62.72 ± 4.37	*89.44 ± 3.08*	71.31 ± 2.27	**87.86 ± 3.37**	**72.18 ± 3.44**	45.58 ± 1.55	**67.70 ± 1.60**
	PPL	14.93 ± 2.91	9.16 ± 0.57	10.39 ± 3.56	13.81 ± 2.60	9.04 ± 1.18	14.35 ± 0.90	15.18 ± 3.72
α = 0.6	Acc	**64.30 ± 6.63**	**90.86 ± 3.24**	71.66 ± 10.09	*87.74 ± 4.04*	71.36 ± 6.25	*47.12 ± 1.66*	66.26 ± 3.24
	PPL	11.49 ± 1.00	7.53 ± 0.68	9.38 ± 0.88	12.15 ± 2.47	7.11 ± 1.08	13.12 ± 0.79	13.29 ± 2.58
α = 0.7	Acc	41.50 ± 13.68	54.52 ± 22.29	50.56 ± 21.95	64.84 ± 16.01	48.12 ± 16.26	38.62 ± 1.59	59.34 ± 7.75
	PPL	8.80 ± 0.42	3.84 ± 0.91	3.43 ± 0.36	4.99 ± 0.28	2.34 ± 0.25	4.28 ± 0.55	6.58 ± 1.40

Table 9: Results with varying α values for Mono-E on Mistral-7B. The best accuracy for each task is in **bold**. For α = 0.4 or 0.6, the model shows outstanding performance on most of the datasets. But increasing alpha to 0.7 results in a nonsensical prompt though the perplexity is super low, leading to corrupted task performance.