State of What Art?
A Call for Multi-Prompt LLM Evaluation

Moran Mizrahi[†] **Guy Kaplan**[†] **Dan Malkin**[†]
Rotem Dror[‡] **Dafna Shahaf**[†] **Gabriel Stanovsky**[†]
[†]School of Computer Science, The Hebrew University of Jerusalem
[‡]Department of Information Systems, University of Haifa
{moranmiz, guykaplan, dan.malkinhueb, dshahaf, gabriel.stanovsky}@cs.huji.ac.il rdror@is.haifa.ac.il

Abstract

Recent advances in LLMs have led to the development of various evaluation benchmarks. These benchmarks typically rely on a *single instruction template* per task. We create a large-scale collection of instruction paraphrases and comprehensively analyze the brittleness of results obtained via single-prompt evaluations across 6.5M instances, involving 20 different LLMs and 39 tasks from 3 benchmarks. We find that different instruction templates lead to very different results, both in terms of absolute performance, as well as relative ranking. Instead, we propose a set of diverse metrics on *multiple instruction paraphrases*, specifically tailored for different use cases (e.g., LLM vs. downstream development), ensuring a more reliable and meaningful assessment of LLM capabilities. We show that our metrics provide new insights into the strengths and limitations of current LLMs.

1 Introduction

Recent years have seen an explosion of large language models (LLMs), which generalize to unseen tasks via natural language instructions. Various LLM evaluation benchmarks, such as BIG-bench and HELM, use a *single* instruction template per task, evaluating all models against it (Srivastava et al., 2022; Liang et al., 2022). However, there could be a myriad of ways to phrase an instruction template for a given task; see Figure 1 for examples of different templates for the task of recognizing homophones. Naturally, LLM performance depends on the chosen template.

In this work, we explore the question of *robustly comparing different models on a given task*. We first create a dataset of paraphrased instructions. To achieve this, we devise three automatic methods to paraphrase given instruction templates, based on recent prompting techniques such as chain-of-thought. We manually verify and filter a large collection of more than 175 paraphrases for different tasks (5K instruction paraphrases in total), which we make publicly available for future research.[1]

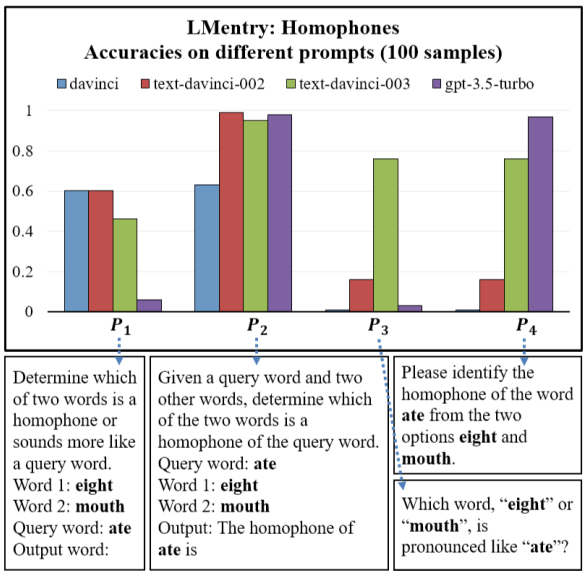

Figure 1: Evaluation of different OpenAI models on the homophones task from LMENTRY over four paraphrases of the instructions for the task. Each cluster of columns corresponds to a distinct *instruction template*, with its respective text detailed below the graph (words in bold indicate a sample-specific instantiation). Despite all instructions being semantically equivalent, both absolute performance and relative ranking vary widely.

Next, we use our dataset to perform a large scale statistical evaluation of over 6.5M instances, involving 20 different LLMs and 39 tasks from 3 benchmarks. We find that models perform very differently on different instruction paraphrases, both in terms of absolute and relative performance. Figure 1 shows an example of the performance of four models on four (semantically equivalent) prompts, with both absolute performance and relative ranking varying widely. At the extreme, there are instruction templates on which a model performs *the best* compared to other models, while on a semantically equivalent instruction the same model

[1] github.com/SLAB-NLP/Multi-Prompt-LLM-Evaluation

performed *the worst* (e.g., GPT-3.5-Turbo on P_1 vs. P_4). Subsequently, we argue that *very little can be said* on either absolute or relative performance based on the common practice of single-instruction evaluation (which may partially explain why some models seem less accurate in practice than their formal evaluation may suggest).

Note that while the claim that evaluating against a single instruction template leads to brittle results is not surprising per se, to the best of our knowledge it has never been subjected to rigorous empirical testing before.

To address the limitations of single-instruction evaluation, we propose to take a step back and consider multi-instruction evaluation metrics which are closely tied to real-world use cases of LLMs. We argue that different use cases should entail different evaluation metrics. For example, LLM developers may be interested in measuring the *robustness of performance* across multiple instruction templates, which we formulate as the average performance across a large collection of instructions. In contrast, when focusing on a downstream task, different models may be better compared according to their corresponding *top-performing* instruction.

We evaluate 20 LLMs with our metrics, finding that their absolute and relative performance differ from those obtained with the benchmarks' original instruction templates. We demonstrate that different models excel in different metrics: For instance, in the LMENTRY benchmark, LLaMA-based models are comparable to T5-based models when looking at top-performing instructions. However, these models lag behind when average performance is considered, due to poor performance on a large number of paraphrases. We also show that our automatic paraphrasing method is effective, and there is no need to manually verify the paraphrases.

Our results suggest that future work should choose the evaluation metric based on the *extrinsic needs* of the evaluators. We hope that our work will help spur more consistency and comparability in LLM evaluation, which is strongly tied to real-world usage of LLMs.

2 Background and Definitions

Below we survey how generalization to a new task format is evaluated and compared between LLMs, finding that this is normally done by testing performance on a single (or very few) task instruction templates. In the rest of the paper, we will argue that such practice leads to brittle results which are not well-suited for real-world use of LLMs.

Task instruction templates. Following Mishra et al. (2021); Chung et al. (2022), we separate between task instruction, samples, and input-output exemplars which may be provided during in-context learning. We define an *instruction template* for a given task as a string with placeholders where the input samples are to be inserted. As seen in Figure 1, the same task can be described using different task instruction templates.

Evaluation benchmarks. Several recent efforts aim to standardize LLM evaluation. Notable examples include MMLU (Hendrycks et al., 2020), BIG-bench (Srivastava et al., 2022; Suzgun et al., 2022), and HELM (Liang et al., 2022). In all of these, each task has a single instruction template, against which all models are evaluated. Another benchmark, LMENTRY (Efrat et al., 2022), reports models' average performance on three instruction templates. The instruction templates are provided with these benchmarks, allowing new models to be tested against the same template.

Sadly, it is also common practice to report results based on a single instruction template *without* making it publicly available (e.g., LLaMA (Touvron et al., 2023), PALM (Chowdhery et al., 2022), GPT-4 (OpenAI, 2023), and Gemini (Google, 2023)). This exacerbates the challenge of meaningful comparative evaluation.

Prompt robustness. Related to this study is a line of work measuring LLM's robustness to prompt (or instruction template) modifications. Unlike our work, these typically aim to measure model performance against *adversarial* paraphrasing approaches. PromptBench (Zhu et al., 2023) measures performance on erroneous instructions (e.g., instructions written by non-native English speakers). They then compare performance on perturbed instructions vs. the benchmark's original instructions, which are considered the gold-standard reference. Gu et al. (2022) examined a single LLM's robustness under various instruction perturbations, including word-, sentence-, and instruction-level changes. Sun et al. (2023) show that LLMs perform better on instructions they have seen in training (BIG-bench Lite benchmark), compared to manual paraphrases. We later incorporate their manual paraphrases in our evaluation.

In contrast to works on prompt robustness, our

scope is wider. We analyze the impact of the choice of prompt in terms of both absolute and relative model performance, covering a wide range of models and several different metrics.

3 Experimental Setup

In this section we describe the tasks and models which we evaluate in this work.

3.1 Tasks

We evaluate 39 diverse tasks from three evaluation benchmarks, as itemized below, and summarized in Table 6 in the Appendix.

10 tasks from LMENTRY (Efrat et al., 2022). LMENTRY consists of simple linguistic tasks (e.g., "write a word that doesn't contain the letter *l*"), each accompanied by three associated instruction templates. The tasks are designed to capture explainable and controllable linguistic phenomena. We choose 10 tasks from LMENTRY that received the lowest scores in the original paper.

14 tasks from BIG-bench Lite (BBL; Srivastava et al., 2022). These cover multiple knowledge domains, sampled from the larger BIG-Bench benchmark (bench authors, 2023). In particular, we focus on a set of 14 tasks studied recently by Sun et al. (2023). Each task in BBL is associated with a single instruction template.

15 tasks from BIG-bench Hard (BBH; Suzgun et al., 2022). This is another curated subset of BIG-bench, containing particularly challenging tasks on which LLM underperform the average human-rater score. We take the set of 15 classification and multiple choice tasks from BBH to ease the evaluation protocol. Each task in BBH is associated with a single instruction template.

Measuring performance. We measure performance in the standard manner provided by each benchmark. In LMENTRY this is done with the official evaluation script, while in Big-Bench we use exact match evaluation. We note that while this evaluation is somewhat strict, we believe that it is also fair and straightforward.

3.2 Models

As shown in Table 1, We evaluate 16 instruction-tuned LLMs from 11 diverse model families (Chung et al., 2022; Sanh et al., 2021; Taori et al., 2023; Zheng et al., 2023; Durbin, 2023; Ding et al., 2023; NousResearch, 2023; Almazrouei et al., 2023; Team, 2023; Collective, 2023). We refrain from including any closed API-based models (e.g., OpenAI models) in our main evaluation for two reasons. First, using them at scale is an expensive prospect, for example, running our entire evaluation suite on GPT-4 will cost up to 2500 USD. Second, and more importantly, the closed API for these models reportedly manipulates the input prompts in an undisclosed manner (e.g., wrapping them with meta-prompts, or rerouting to other models) (Rao et al., 2023) which interferes with our evaluation. We do however perform a small-scale evaluation of OpenAI models in Section 7 to show that they are also sensitive to prompt paraphrasing.

Model	Size	Base Model	# Params
Flan-T5	Small	T5	80M
	Base		250M
	Large		780M
	XL		3B
	XXL		11B
T0	Small	T5	3B
	T0pp		11B
Alpaca	Small	LLaMA	7B
	Big		13B
Vicuna		LLaMA	13B
Airoboros		LLaMA	13B
UltraLM		LLaMA	13B
Nous-Hermes		LLaMA	13B
Falcon-Instruct		Falcon	7B
MPT		MPT	7B
Minotaur		StarCoder Plus	15B

Table 1: The different LLMs evaluated in this work, grouped by model family, along with their size, in number of parameters. All models were instruction-tuned.

4 Single-Prompt Evaluation Leads to Inconsistent Results

As discussed in the previous section, a common practice in LLM evaluation is to evaluate different models against a single instruction template. In this section, we will show that this approach is quite brittle. Indeed, a simple rephrasing of the instruction template can lead to drastic changes in absolute model performance as well as its relative ranking among other models.

To show this, in Section 4.1 we create a large number of instruction paraphrases for each of our tasks. This is achieved automatically with the aid of an LLM and verified by human annotators to

Benchmark	Method	#Automatic Paraphrases	#Correct Paraphrases	Correct Ratio
LMENTRY	All	2429	2186	90.00%
	Rephrase	461	408	88.50%
	CoT	1286	1234	95.96%
	Gradual	652	514	78.83%
BBH	All	2615	2209	84.47%
	Rephrase	734	627	85.42%
	CoT	775	630	81.29%
	Gradual	1091	937	85.88%

Table 2: Manual validation and filtering of automatic instruction paraphrases generated for LMENTRY and BBH, showing percentages of valid paraphrases.

reduce noise. Then, in Section 4.2, we statistically analyze the performance of various LLMs against these instruction templates and quantify the variation in model performance and ranking.

4.1 Paraphrasing Instruction Templates

We use three prompting methods which were found useful in previous work: (1) instruction template rephrasing: asking an LLM to rephrase a seed prompt (Lester et al., 2021; Gonen et al., 2022; Honovich et al., 2022a); (2) Chain-of-Thought prompting (Wei et al., 2022): we provided the model with a sequence of steps in which the model is asked first to produce a task description, and then to generate various instruction templates for the task; and (3) Gradual template generation: inspired by Honovich et al. (2022b), we split the COT approach into three LLM calls. The first for generating a task description from a seed instruction template, the second for generating instruction provided by input-output examples, and the third for processing the instruction and examples into an instruction template. See more details about these approaches in the Appendix.

We use the original instruction templates for each of our tasks to seed these three generation methods, resulting on average in more than 200 automatically-generated instruction template paraphrases for each of our tasks (see Table 2). We make this collection, as well as the code used to generate it, publicly available for reproducibility and to enable future work.

Manual validation and filtering of automatic instruction paraphrases. We manually verify and filter all of the automatically generated paraphrases.We found that 90% of the generated paraphrases created for LMENTRY were correct, and roughly 84% of the paraphrases for BBH were correct. See Table 2 for a fine-grained distribution across the different generation metrics. On average, this process yields more than 175 validated instruction paraphrases per task across LMENTRY and BBH, which we will subsequently use to quantify peformance variability due to instruction template paraphrasing.

4.2 Quantifying Performance Variance due to Instruction Paraphrasing

We leverage the collection of validated instruction paraphrases to show that model performance varies widely on different instruction templates, both at the individual model performance, as well as in relative model ranking. As we argue below, our main finding is that the common approach of evaluating against a single instruction template is inconsistent and unstable, leading to contradicting results.

Instance sampling and prompt construction. Evaluating LLMs can become prohibitively expensive with the increase of the number of samples, datasets, models, and instruction templates (Perlitz et al., 2023). We focus on a large number of tasks, models, and instruction paraphrases. Hence, to make our evaluation feasible, this comes at the expense of the number of samples per task. Concretely, we evaluate each instruction template on a randomly selected subset of 100 task samples. Furthermore, we found that all models struggle on BBH, beyond the point of meaningful comparison. To address this, we evaluate 11 out of the 16 models on it (the bigger ones in terms of number of parameters), and we add an example of the prediction format to all instruction template paraphrases. Examining the effect of few-shot learning is beyond the scope of this paper, however, Sclar et al. (2023) recently observed similar performance sensibility when introducing varying number of in-context examples.

Using a single-instruction template leads to brittle ranking. We compute Kendall's $W : \mathbb{N}^{m \times n} \mapsto [0, 1]$ (Kendall and Smith, 1939), a non-parametric statistic which measures the ranking correlation between m judges (instruction templates, in our case) ranking n objects (LLMs, in our case) by calculating the squared deviation between the sum of ranks of different judges ($R_i = \sum_{j=1}^{m} r_{ij}$) and their mean value:

$$W = \frac{12 \sum_{i=1}^{n} (R_i - \bar{R})^2}{m^2 (n^3 - n)}$$

Kendall's W would be 1 for all tasks if model

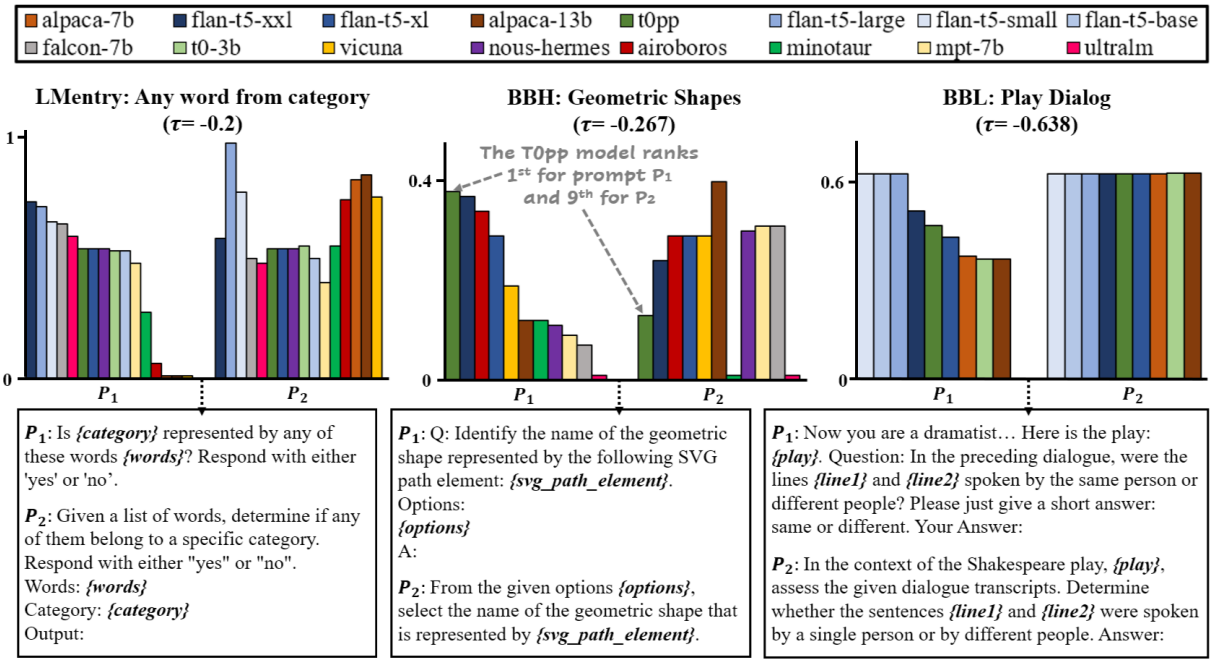

Figure 2: Model performance and ranking induced by pairs of instruction templates that exhibit the minimal Kendall τ correlation on three different tasks (one for each benchmark). Models are consistently ordered across graphs to ease comparison of the ranking changes between each template pair.

ranking were the same among all instruction templates (in other words, they are interchangeable for the sake of evaluation). In contrast, the more W approaches 0, the lesser the rankings induced by different instructions agree.

The results (Table 3) demonstrate that a single instruction template leads to unreliable rankings for many of the tasks, with 10 of the tasks exhibiting only slight to moderate ranking agreement, and only two exhibiting strong agreement. To complement the analysis, we performed Friedman test with tied data (Corder and Foreman, 2011), showing that different instructions lead to statistically significant differences in performance for 21 out of the 25 tasks.

Examples of differences in model ranking. We illustrate the implications of such differences in Figure 2. The three instruction template pairs are valid paraphrases, yet they lead to vastly different results. For example, T0pp ranks first on the BBH task using the first instruction template and only 9th using the second template. Similarly, Alpaca-13B and Alpaca-7B are in the *top* performing models on the LMENTRY task using the second instruction template, while they rank *last* in the first template.

We quantify the difference between two rankings with Kendall's $\tau : \mathbb{N}^n \times \mathbb{N}^n \mapsto [-1, 1]$, which estimates the agreement between two specific instruction templates which induce rankings R_1, R_2 over n LLMs, formally defined as (Kendall, 1945):

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + T) \cdot (P + Q + U)}}$$

Where P is the number of concordant pairs, Q is the number of discordant pairs, T is the number of ties in the first ranking, and U is the number of ties in the second ranking. Therefore, $\tau > 0$ indicates that most pairs are concordant (with $\tau = 1$ indicating perfect agreement), and $\tau < 0$ indicates that most pairs are discordant (with $\tau = -1$ indicating perfect disagreement).

Appendix A.4 presents examples of pairs of instruction templates that exhibit the minimal Kendall τ correlation per task (i.e., their ranking is most dissimilar). Overall, 15 tasks have instruction template paraphrases with negative Kendall's τ, indicating mostly disagreeing LLM rankings.

Absolute model performance varies widely on single-instruction templates. Aside from vastly different relative model rankings, instruction template paraphrases often result in varying absolute model performances. To quantify this variance, we calculated *divergence*, defined as the number of standard deviations by which the performance, as assessed using the original instruction templates,

Tasks	Kendall's W	Friedman p
LMENTRY		
not containing	.271 (weak)	**0.0***
word before	.367 (weak)	**0.0***
first alphabet	.436 (weak)	**0.0***
less letters	.485 (weak)	**0.0***
rhyming word	.496 (weak)	**0.0***
ends with word	.518 (weak)	**0.0***
homophones	.518 (weak)	**0.0***
all words	.522 (weak)	**0.0***
any words	.527 (weak)	**0.0***
more letters	.540 (weak)	**0.0***
BIG-bench Hard		
recommendations	.628 (medium)	.897
formal fallacies	.704 (medium)	**5.6E-13**
geometric shapes	.710 (medium)	.167
hyperbaton	.730 (medium)	**1.0E-4**
logical deduction 3	.740 (medium)	**4.9E-16**
disambiguation qa	.764 (medium)	**2.1E-17**
ruin names	.776 (medium)	.366
logical deduction 7	.778 (medium)	**1.4E-13**
translation error	.800 (medium)	**6.9E-9**
logical deduction 5	.818 (medium)	**3.0E-9**
snarks	.823 (medium)	.604
penguins in a table	.830 (medium)	**7.3E-15**
navigate	.838 (medium)	**5.6E-10**
causal judgement	.851 (strong)	**4.9E-7**
sports	.873 (strong)	**8.0E-13**

Table 3: Kendall's $W \in [0, 1]$ values for all tasks sorted in ascending order. The smaller the value of W the more that the ranking on different prompts is de-correlated. Most W are smaller than 0.85, indicating weak to moderate agreement. The p-values from Friedman test indicate significant differences between rankings of models when using different prompts. *p-values of 0.0 represent statistical significance levels that are smaller than 1E-50.

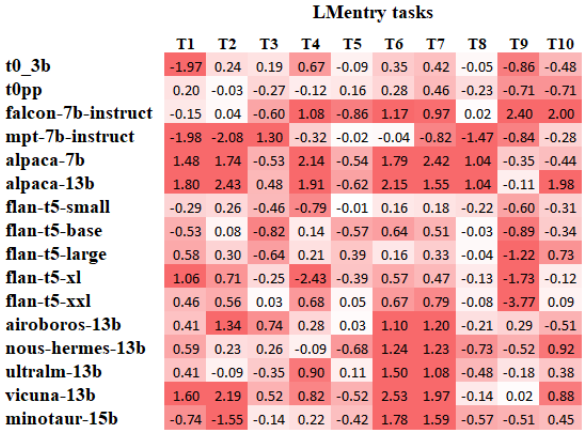

Figure 3: Model and task performance divergence, showing for each task in LMENTRY the number of standard deviations by which the performance of each model on the original instruction templates deviates from the averaged model performance. Dark red cells indicate substantial divergence values exceeding one standard deviation.

deviates from the model's average performance over all paraphrases.

The results in Figure 3 reveal noticeable divergence for the LMENTRY benchmark, defined as surpassing one standard deviation (Kazmier et al., 2003). For instance, the performance of the Alpaca-13B with the original instruction templates outperformed its average performance by more than one standard deviation in 7 out of the 10 LMENTRY tasks. For lack of space, the figure does not depict the BBH benchmark, but similar patterns of divergence were observed there as well.

In line with Lou et al. (2023), we find that major differences in performance can occur even for very similar paraphrase pairs. For example, the Flan-T5-large model demonstrated an average performance degradation of 28% when changing the word 'excludes' to 'lacks', while the Flan-T5-XL model showed an average performance improvement of 46% on that same edit. See a comprehensive edit distance comparison in Appendix A.5.

4.3 LLMs are also Sensitive to Manual Paraphrases

It is possible that the inconsistencies observed in our analyses stem from our automatic paraphrases. To address this, we extended our analysis with instruction paraphrases which were recently written by Sun et al. (2023) for the BBL tasks (see Table 6). These provide between 7 and 12 instruction templates per task. While originally annotated to examine overall model degradation on human written instructions, we reuse Sun et al. (2023)'s annotations to examine the change in model rankings and absolute performance.

Our analysis revealed similar inconsistencies as observed with automated paraphrases. See Table 13 in the Appendix for the Kendall's W values for all BBL tasks, and Table 11 for examples of pairs of instruction templates that exhibit the minimal Kendall τ correlations.

5 Different Use Cases Merit Different Metrics

So far we have shown that LLM performance is greatly affected by paraphrasing of instruction templates. This calls into question current evaluation practices, which typically rely on LLM performance on a single instruction template. In this section we explore ways to evaluate LLMs using a *diverse set of instruction templates*.

Most importantly, we argue that the answer should depend on the *purpose of the evaluation*, and that different extrinsic needs should lead to different evaluation metrics, rather than striving for a coarse catch-all metric. We introduce a set of metrics, each tailored to specific scenarios and realistic user needs.

Notations. In the following, M is a pretrained LLM, $T = \{(x_i, y_i)\}$ denotes an evaluation dataset for M, I_T is a set of natural language task instruction paraphrases for T (e.g., obtained via automatic paraphrasing), and $\varepsilon(M, T, i) \in [0, 1]$ denotes the aggregated performance of M on samples from T, using a single instruction template $i \in I_T$ according to a standard metric, e.g., accuracy or F_1.

5.1 Maximum Performance Metric – For Particular Downstream Applications

We define the maximum performance (MaxP) of a model M on task T to be the maximum individual instruction template performance this model achieves across all instruction templates:

$$MaxP(M, T, I_T) = \max_{i \in I_T} \varepsilon(M, T, i)$$

Use case: This metric is useful for developers aiming to integrate an LLM into a specific downstream task and domain (for example, sentiment analysis in the news domain). In such cases, a user input is often embedded within a fixed instruction template. As such, it makes sense to find the best-performing instruction template for a given model (Wei et al., 2021). To mitigate overfitting, it is sensible to identify it using a held-out sample.

5.2 Average Performance Metric – For LLM Developers

We define the average performance (AvgP) of a model M on task T as the mean of the individual instruction template performances over all instruction templates for the task:

$$AvgP(M, T, I_T) = \frac{1}{|I_T|} \cdot \sum_{i \in I_T} \varepsilon(M, T, i)$$

Use case: Average prompt performance is useful for assessing model robustness to paraphrases. We believe this should be standard practice for LLM developers when presenting the performance of a new LLM on a range of tasks and prompt paraphrases (Workshop et al., 2022), as it mitigates outliers in performance.

Benchmark	MaxP	AvgP	Combined
LMENTRY	.963	.978	.948
BBH	.991	.983	.966

Table 4: Averaged Kendall's Tau values comparing rankings before and after filtering incorrect paraphrases for each metric across all tasks (excluding "ends with word" for LMENTRY).

5.3 Combined Performance Score

In the same way the F1 score combines precision and recall into a single metric, we propose a Combined Performance Score (CPS) that unites the maximum and average performance metrics to capture both peak capability and consistency of the model across prompts. To define CPS, we first introduce a model saturation score:

$$Sat(M, T, I_T) = 1 - (MaxP - AvgP)$$

This score measures how closely the model's best performance aligns with its average performance. A high saturation score indicates that the model's performance does not drop significantly for non-optimal instructions. Then, the CPS is calculated as the product of the model's best performance ($MaxP$) and its saturation (Sat):

$$CPS(M, T, I_T) = Sat \cdot MaxP$$

Use case: This metric is valuable for selecting a model for a suite of applications or a platform offering diverse tasks. For instance, when integrating an LLM into an application with user-visible prompts, such as a multi-functional chatbot, it is crucial for the model to be both effective (high $MaxP$) and consistent (high Sat). CPS facilitates identifying models that strike a balance between top-tier performance and consistent reliability across varying instruction templates.

6 Multi-Prompt Evaluation

In Figure 4 we evaluate all our 16 models according to the metrics we proposed in the previous section, on sample tasks from each of the three benchmarks (full results for all tasks are available in our repository). We report several interesting observations.

First, we find that all aggregate metrics diverge from the performance on the original instruction templates. For the vast majority of the tasks in our study, the top three models determined by the

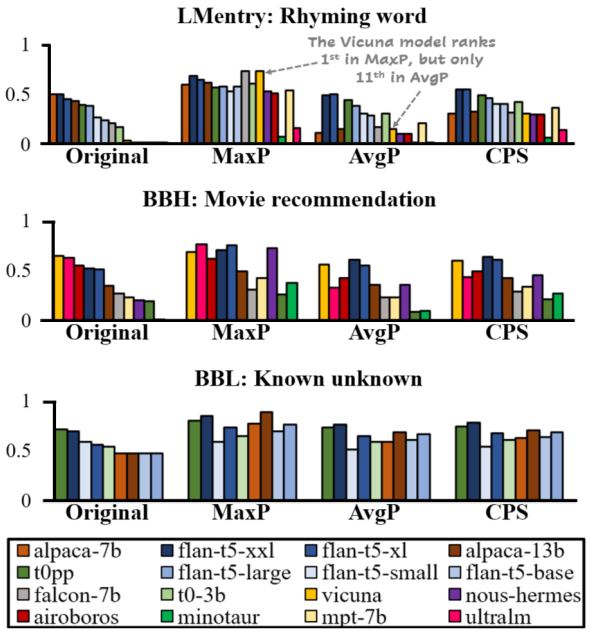

Figure 4: The performance of various models according to the metrics proposed in Section 4, evaluated on sample tasks from each of the three benchmarks. The name of the metric appears below each group of columns; height of a column represents value in *that specific metric*. The order of the columns (i.e., models) between groups is fixed, set according to decreasing performance on the original instruction templates.

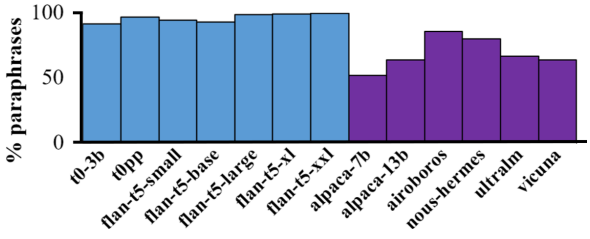

Figure 5: Percentage of correct paraphrases with accuracy higher than 5% in T5 models (blue) vs. LLaMA models (purple) on LMENTRY tasks.

original instruction templates were different from those which ranked first according to the average and maximum metrics.

More broadly, the rankings of models depend on the metric used. For instance, see Figure 4 (top): In LMENTRY's rhyming word task, Falcon-Instruct-7b and Vicuna-13b rank first according to $MaxP$ (0.74, gray and yellow bars), but their average performances $AvgP$ are only 0.17 and 0.15, respectively. Similarly, across all tasks in the LMENTRY benchmark, LLaMA-based models were competitive with T5-based models in terms of $MaxP$. However, in terms of $AvgP$, they tended to lag behind, due to extremely poor performance on a large number of paraphrases (see Figure 5 for percentage of paraphrases that achieved over 5% accuracy).

Finally, we found that noise stemming from automatic paraphrase generation has virtually no impact on metric-based model rankings. We compute Kendall's τ to compare model rankings before and after the manual removal of incorrect paraphrases. The results (Table 4) show near-perfect to perfect agreement in rankings across all tasks, except for the "ends with word" task in LMENTRY. Upon examination, this seems to be mostly due to an error in LMENTRY's evaluation script. These results suggest that it may be enough to compute our metrics over range of automatically-generated paraphrases, without having to manually verify them.

7 Small-Scale Evaluation of OpenAI Models on Prompt Paraphrasing

In this section we perform a small-scale evaluation showing that API LLMs are also sensitive to instruction paraphrasing. Our evaluation focuses on four OpenAI models: davinci, text-davinci-002, text-davinci-003, and GPT-3.5-Turbo on the LMENTRY benchmark.

Due to budget constraints, we show that the performance of these models diverges significantly between the benchmark's original instruction templates and a selection of paraphrases, in terms of both average and maximum metrics.

Estimating average performance. To estimate the average performance of OpenAI models on a specific task, we adopted a randomized approach. For each task sample, we randomly selected a paraphrase from our collection, and evaluated the model's response, scoring the entire set of task samples. To approximate average performance, this experiment was repeated 20 times, determined by the data from our 16 open-source models.

Estimating maximal performance. To estimate which of the roughly 175 instruction templates per task performs the best for each model, we implemented a simple greedy search. Initially, we evaluated all paraphrases on 10 task instances, then narrowed down to the top 100 instruction templates for another 10 instances. Finally, the top 10 instruction templates were evaluated on the remaining instances, and the template that performed the best was chosen to estimate the maximum performance.

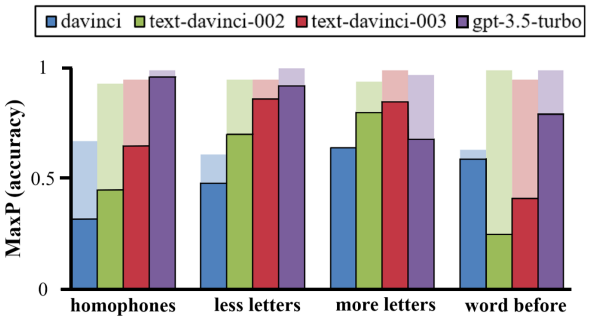

Figure 6: Comparison of the *maximum performance* of four OpenAI models using original prompts (in solid colors) vs. all prompt paraphrases (semi-transparent). Each group of columns corresponds to a different task in the LMENTRY benchmark.

7.1 Results

Below we summarize the results of our evaluation of OpenAI models. The full details appear in Tables 21, 22, 23, and 24 and in our repository.[1]

OpenAI models are also sensitive to minor prompt variations. Minor changes in the phrasing of the instruction can lead to drastic performance changes for the OpenAI models in our experiment, similar to our findings in Section 4.2 with smaller-scale LLMs. See representative examples in Table 5, showing nearly identical instruction template pairs resulting in notable variations in performance.

Average multi-prompt performance is lower than that observed in the original benchmark instructions. In 72.5% of the cases, the performance of the original instruction templates was higher than the estimated average across all paraphrases. A prominent difference was observed particularly in the davinci model. For this model, the original prompts added, on average, 21 more accuracy points compared to the estimated average across all paraphrases.

Original prompt performances fall below all paraphrases' estimated maximum performance. Figure 6 depicts maximum performance of the *original instructions* for four LMENTRY tasks in solid colors, with overlaid semi-transparent columns indicating the estimated maximum performance on *all paraphrases*. Notably, for text-davinci-002, we found paraphrases that improved its maximal accuracy performance above 90% for 8 out of 10 tasks. Across all four models, 26 out of 40 differences were statistically significant according to the McNemar test (Table 25).

Model rankings diverge between the different metrics and original instruction templates. Similarly to our main evaluation, there were many mismatches between ranking on the original instruction templates and our metrics. Agreement was observed in only 5 out of 10 tasks for the average metric, and in 4 out of 10 tasks for the maximum metric.

8 Related Work

Our work is part of an emerging trend highlighting the many challenges standing in the way of meaningful, scalable, and reproducible evaluation of large language models.

Perlitz et al. (2023) focus on the rising cost of exhaustive evaluation of LLMs on large number of samples. They notice that as models become larger, the cost of running them at scale can become prohibitively expensive, even during inference. To help mitigate this problem, they develop methods for choosing subsets of the test data which are expected to be a good representative of the whole. We find that single prompt evaluation is not a good representative of LLMs average performance, and instead suggest evaluating on many instruction templates per sample, which further increases the evaluation cost. An interesting avenue for future work can extend Perlitz et al. (2023)'s approach to also include various instruction templates, thus efficiently approximating our suggested evaluation methods.

Sclar et al. (2023) show that LLMs are sensitive to *prompt formatting*. These are minor prompt design choices, such as the addition or omission of punctuation marks. They create a large pool of instruction paraphrases, ensuring that paraphrases maintain the meaning of the original prompt. We notice a similar phenomenon, albeit more anecdotally, when our automatic paraphrasing techniques incidentally produce minor changes in formatting (Table 5). Finally, Voronov et al. (2024) shows that LLMs are sensitive to how in-context examples are presented and formatted. For example, they vary the manner in which each input-output is separated, and test how such choices interact with the phrasing of the instruction template, the number of demonstrations, or the model size.

Our work distinguishes itself as the first to systematically explore the impact of a broad spectrum of prompt paraphrases across various benchmarks

Change	Model	P1	Acc.	P2	Acc.	Diff.
{...} –> "{...}"	td002	Which word has a greater number of letters, ***{word1}*** or ***{word2}***?	.50	Which word has a greater number of letters, ***"{word1}"*** or ***"{word2}"***?	.23	-0.27
	td002	Which of the words ***{word1}*** and ***{word2}*** is alphabetically first?	.54	Which of the words ***"{word1}"*** and ***"{word2}"*** is alphabetically first?	.77	+0.23
	td003	Which word has a greater number of letters, ***{word1}*** or ***{word2}***?	.60	Which word has a greater number of letters, ***"{word1}"*** or ***"{word2}"***?	.14	-0.46
	td003	Compare the length of ***{word1}*** and ***{word2}*** and tell me which one is shorter.	.39	Compare the length of ***"{word1}"*** and ***"{word2}"*** and tell me which one is shorter.	.73	+0.34
	cgpt	Which word has a greater number of letters, ***{word1}*** or ***{word2}***?	.55	Which word has a greater number of letters, ***"{word1}"*** or ***"{word2}"***?	.24	-0.31
	cgpt	Compare the length of ***{word1}*** and ***{word2}***. Which one is longer?	.04	Compare the length of ***"{word1}"*** and ***"{word2}"***. Which one is longer?	.70	+0.66
',' –> ':'	td002	Which word is a rhyme for "{query}**,**" "{word1}" or "{word2}"?	.08	Which word is a rhyme for "{query}**:**" "{word1}" or "{word2}"?	.85	+0.77
	td003	Which word is a rhyme for "{query}**,**" "{word1}" or "{word2}"?	.48	Which word is a rhyme for "{query}**:**" "{word1}" or "{word2}"?	.90	+0.42
',' –> '-'	td002	Which word rhymes with "{query}**,**" "{word1}" or "{word2}"?	.06	Which word rhymes with "{query}" **-** "{word1}" or "{word2}"?	.73	+0.67
	td003	Which word rhymes with "{query}**,**" "{word1}" or "{word2}"?	.17	Which word rhymes with "{query}" **-** "{word1}" or "{word2}"?	.60	+0.43
the –> a	td002	What is ***the*** word that rhymes with "{query}" - "{word1}" or "{word2}"?	.03	What is ***a*** word that rhymes with "{query}" - "{word1}" or "{word2}"?	.78	+0.75
which –> what	td002	***Which*** word rhymes with "{query}" - "{word1}" or "{word2}"?	.73	***What*** word rhymes with "{query}" - "{word1}" or "{word2}"?	.82	+0.09
	td003	***Which*** word rhymes with "{query}" - "{word1}" or "{word2}"?	.60	***What*** word rhymes with "{query}" - "{word1}" or "{word2}"?	.15	-0.45
word –> term	td002	Create a ***word*** that excludes the letter "{letter}".	.54	Create a ***term*** that excludes the letter "{letter}".	.04	-0.50
	td003	Create a ***word*** that excludes the letter "{letter}".	.96	Create a ***term*** that excludes the letter "{letter}".	.58	-0.38
	cgpt	Create a ***word*** that excludes the letter "{letter}".	.81	Create a ***term*** that excludes the letter "{letter}".	.42	-0.39

Table 5: Minimal distance paraphrase pairs from LMENTRY with large performance differences in OpenAI models.

and tasks on multiple models, coupled with a statistical analysis of the absolute and relative variations in evaluations. Furthermore, we introduce a suite of metrics specifically designed to align with the practical applications of large language models.

9 Conclusions

Our research highlights the sensitivity of large language models (LLMs) to prompt paraphrasing, challenging the adequacy of single-prompt evaluations. We propose alternative evaluation metrics that use a diverse set of instruction templates for each task, designed for more robust and meaningful LLM evaluation. For example, LLM developers may be interested in measuring the robustness of performance across multiple prompts, which we propose to evaluate as the average across a large collection of prompts. In contrast, when developing a downstream model, different models should be compared according to their corresponding top-performing prompt.

Evaluating based on these metrics underscores the necessity for nuanced evaluation methods, revealing notable differences in absolute performance and relative model rankings compared to traditional evaluations. We hope that our work will help spur more consistency and comparability in LLM evaluation which is strongly coupled to real-world LLM uses. We believe this shift is crucial for accurately understanding and leveraging the true capabilities of LLMs.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, et al. 2023. Falcon-40b: an open large language model with state-of-the-art performance. Technical report, Technical report, Technology Innovation Institute.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of lan-

guage models. *Transactions on Machine Learning Research*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

OpenAccess AI Collective. 2023. Minotaur.

Gregory W Corder and Dale I Foreman. 2011. Nonparametric statistics for non-statisticians.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations.

Jon Durbin. 2023. Airoboros.

Avia Efrat, Or Honovich, and Omer Levy. 2022. Lmentry: A language model benchmark of elementary language tasks. *arXiv preprint arXiv:2211.02069*.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models.

Jiasheng Gu, Hanzi Xu, Liangyu Nie, and Wenpeng Yin. 2022. Robustness of learning from task instructions. *arXiv preprint arXiv:2212.03813*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022a. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2022b. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*.

Leonard J Kazmier, Michael K Staton, Daniel L Fulks, et al. 2003. Business statistics: based on schaums outline of theory and problems of business statistics, by leonard j. kazmier. Technical report, McGraw-Hill.

Maurice G Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Maurice G Kendall and B Babington Smith. 1939. The problem of m rankings. *The annals of mathematical statistics*, 10(3):275–287.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2023. Is prompt all you need? no. a comprehensive and broader view of instruction learning. *arXiv preprint arXiv:2303.10475*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

NousResearch. 2023. Nous-hermes.

OpenAI. 2023. Gpt-4 technical report.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2023. Efficient benchmarking (of language models). *ArXiv*, abs/2308.11696.

Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. 2023. Tricking llms into disobedience: Understanding, analyzing, and preventing jailbreaks. *ArXiv*, abs/2305.14965.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *ArXiv*, abs/2310.11324.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2023. Evaluating the zero-shot robustness of instruction-tuned language models. *arXiv preprint arXiv:2306.11270*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

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

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

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

A Appendix

A.1 Tasks - Additional Details

Table 6 presents an overview of the 39 tasks from the 3 benchmarks discussed in this paper: LMENTRY, BIG-bench Lite, and BIG-bench Hard. These benchmarks include 10, 14, and 15 tasks from each, respectively. The table also provides an example task instruction for each task.

A.2 Process of Generating Prompt Paraphrases

Our process for generating paraphrases of instruction templates is depicted with an example in Figure 7.

A.3 Paraphrases Correctness

Tables 7 and 8 present the percentages of correct paraphrases that were generated by the 3 prompt-generating methods presented in the paper for LMENTRY and BBH. The tables also depict the average model accuracy and standard deviations as measured for only the correct paraphrases across all LLMs. The correct paraphrases were identified by one of the authors of this paper. Table 14 presents the Kendall τ values before and after the removal of incorrect paraphrases. The agreement in the ranking of models is near-perfect to perfect in both LMENTRY and BBH benchmarks.

A.4 Comparing Different Instruction Templates with Kendall's τ Rank Disagreements

Tables 9 , 11, and 10 present the Kendall τ values of representative examples from all benchmarks with Kendall τ values that are significantly different from 0. i.e., notable variations in rankings of models for two paraphrases of the same task instruction.

A.5 Model Performance Differences with Minimal Paraphrasing Edit Distance

Figure 8 depicts the average performance differences between various LLMs when small edits are made to the instruction templates.

In addition, Table 12 shows representative examples of instruction template pairs with very minor differences but notable variations in performance.

A.6 BBL Analysis

This subsection consists of an additional analysis of the BBL benchmark that was not detailed in the main body of the paper. Table 3 presents the Kendall's W values and the Friedman test p-values that demonstrate a low correlation between the ranks of the models for different instruction templates and reveal similar inconsistencies as observed with automated paraphrases in other benchmarks. Figure 10 shows the deviation of the original instruction template from the average performance calculated over the generated instruction templates of several models for all of the BBL tasks.

A.7 Average Model Ranks for Each Metric Across All Tasks

Tables 15, 16 present the average model ranks for each metric across all tasks in LMENTRY and BBH respectively. Flan-T5-XXL emerges as the top performer for all metrics in both benchmarks. Minotaur is at the bottom of the performance spectrum across all evaluated models in BBH.

A.8 Analysis of Origin Generation Method of Optimal Paraphrases

Our analyses for the origin of the optimal paraphrases used by each model, are summarized in Tables 17, 18. The gradual method surfaced as the dominant source of optimal paraphrases across both benchmarks, particularly pronounced in the LMENTRY benchmark. However, a closer look at individual models revealed a pattern of preference for different generation methods.

A.9 Small Scale Evaluation - OpenAI

This subsection contains all the tables referenced in Section 7. Table 19 and Table 20 are related to our naive heuristics for estimating average and maximum performance, respectively. Table 19 presents the average number of repetitions needed for our heuristic to estimate the average performance, ensuring less than a 1-point accuracy discrepancy from the actual average for each open-source model across all tasks in the LMENTRY benchmark. Table 20 compiles results from our greedy heuristic that searches for the optimal paraphrases for each open-source model on each LMENTRY task.

Table 21 and Table 23 aggregate the average and maximum performances for each model and task using only the original instruction templates. Similarly, Table 22 and Table 24 present the approximated average and maximum performances, computed with our heuristics, for each model and task using all paraphrased templates.

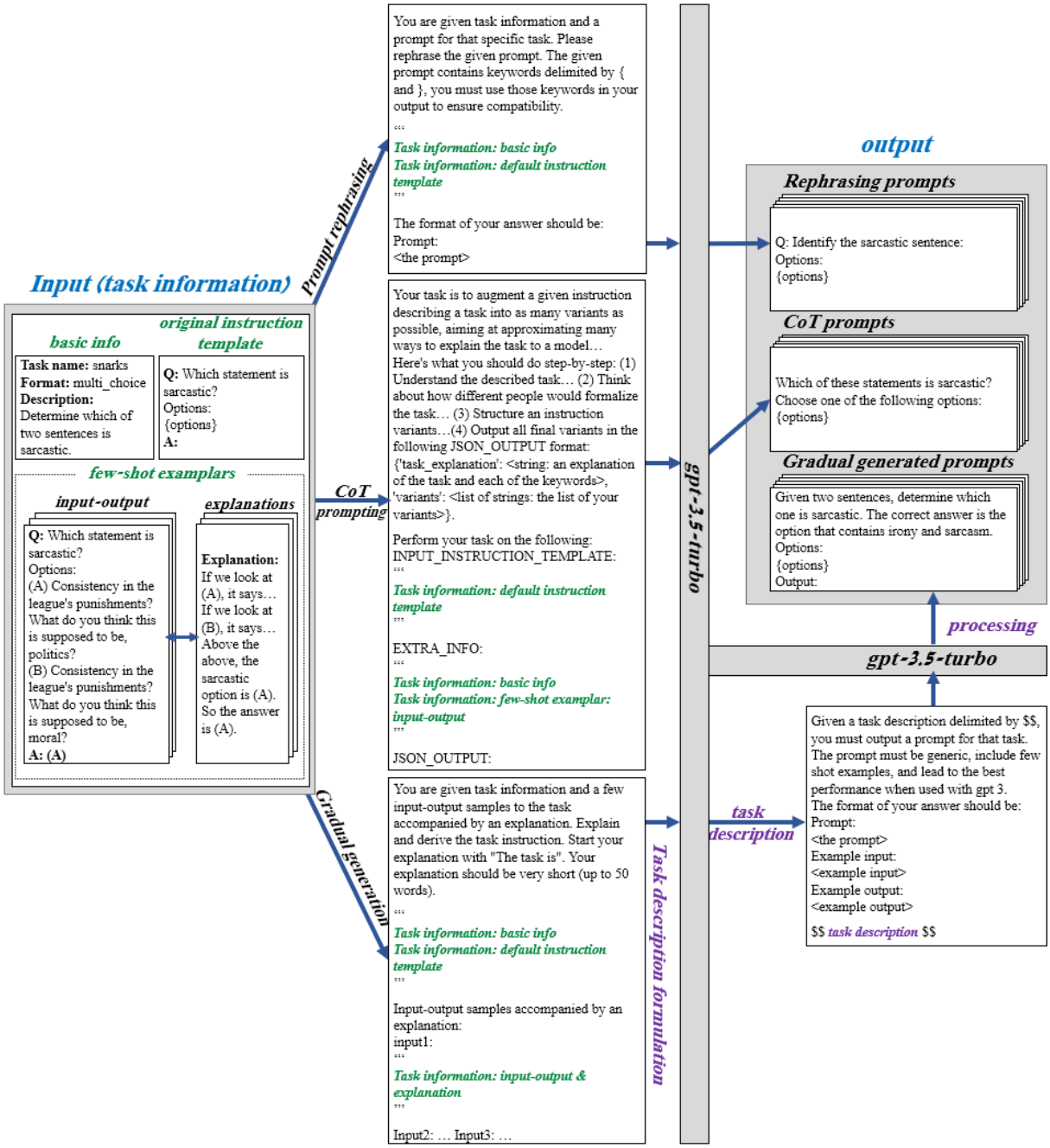

Figure 7: Our process for automatically generating paraphrases of instruction templates, using the 'snarks' task from the BBH benchmark as an example. We input task information from the benchmark, including basic details, the original instruction template, and a few-shot exemplar, into various meta-prompts tailored to different generation methods (prompt rephrasing, CoT prompting, or gradual generation). Then, we feed these meta-prompts into gpt-3.5-turbo to create new instruction templates for the given task. Notably, in the gradual generation method, gpt-3.5-turbo is utilized twice: initially to generate a detailed task description, and subsequently to derive a new instruction template from it.

Benchmark & Task	Instruction Template
LMENTRY	
all words from category	Q: Are all the words {$words$} types of {$category$}? Answer either "yes" or "no". A:
any words from category	Q: Does the list [{$words$}] contain any {$category$}? Answer either "yes" or "no". A:
ends with word	Write a sentence that ends with the word "{$word$}":
first alphabetically	Q: In an alphabetical order, which word comes first, "{$word1$}" or "{$word2$}"? A:
homophones	Q: Which word sounds like the word "{$query$}", "{$word1$}" or "{$word2$}"? A:
less letters	Q: Which word is shorter, "{$word1$}" or "{$word2$}"? A:
more letters	Q: Which word has more letters, "{$word1$}" or "{$word2$}"? A:
rhyming word	Q: Which is a rhyme of the word "{$query$}", "{$word1$}" or "{$word2$}"? A:
word before	Q: Which word comes right before "{$word$}" in the sentence "{$sentence$}"? A:
word not containing	Write a word that doesn't contain the letter "{$letter$}":
BIG-bench Lite	
bbq lite	{$input$} option: {$option1$} option: {$option2$} option: {$option3$} Answer:
code line description	Python code: {$input$} choice: {$option1$} choice: {$option2$} choice: {$option3$} choice: {$option4$} English language description:
conceptual combinations	{$input$} option: {$option1$} option: {$option2$} option: {$option3$} option: {$option4$} Answer:
hindu knowledge	Q: {$input$} choice: {$option1$} choice: {$option2$} choice: {$option3$} choice: {$option4$} A:
known unknown	{$input$} choice: {$option1$} choice: {$option2$}
language identification	Given a sentence, select the correct language among the choices Sentence: {$input$} choice: {$option1$} choice: {$option2$} choice: {$option3$} choice: {$option4$} ... Language:
logic grid puzzle	{$input$} Answer:
logical deduction	The following paragraphs each describe a set of three objects arranged in a fixed order. The statements are logically consistent within each paragraph. {$input$}
novel concepts	Let's do some find-the-common-concept problems. In these problems, your goal is to identify the underlying concept or theme that relates the things listed. Make sure to answer carefully. {$input$} Answer:
play dialog	The following transcripts of dialogues have been taken from Shakespeare plays, but the transcripts do not say who said what. Your task is to identify whether the sentences in question were spoken by the same or different people. Dialogue: {$input$} Answer:
strange stories	Context: {$input$} choice: {$option1$} choice: {$option2$} choice: {$option3$} choice: {$option4$} A:
strategic qa	Q: {$input$} A:
vitaminc fact verification	Based only on the information contained in a brief quote from Wikipedia, answer whether the related claim is True, False or Neither. Use Neither when the Wikipedia quote does not provide the necessary information to resolve the question. Passage: {$input$} True, False, or Neither?
winowhy	Please answer the following questions about which words certain pronouns refer to. {$input$} The above reasoning is
BIG-bench Hard	
causal judgement	How would a typical person answer each of the following questions about causation?
disambiguation qa	In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.
formal fallacies	Is the argument, given the explicitly stated premises, deductively valid or invalid?
geometric shapes	This SVG path element {$svg\_path\_element$} draws a Options: {$options$}
hyperbaton	Which sentence has the correct adjective order:
logical deduction five objects	QThe following paragraphs each describe a set of five objects arranged in a fixed order. The statements are logically consistent within each paragraph.
logical deduction seven objects	The following paragraphs each describe a set of seven objects arranged in a fixed order. The statements are logically consistent within each paragraph
logical deduction three objects	The following paragraphs each describe a set of three objects arranged in a fixed order. The statements are logically consistent within each paragraph.
movie recommendation	Find a movie similar to {$movie\_list$}
navigate	If you follow these instructions, do you return to the starting point?
penguins in a table	Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) {$question$}
ruin names	Q: Which of the following is a humorous edit of this artist or movie name: '{$artist\_or\_movie\_name$}'?
salient translation error detection	The following translations from German to English contain a particular error. That error will be one of the following types: . . . Please identify that error.
snarks	Which statement is sarcastic?
sports understanding	Q: Is the following sentence plausible?

Table 6: The 39 tasks used in this paper, along with the benchmarks from which they were taken and an example task instruction.

Benchmark & Task	Method	#Auto Paraphrases	#Correct Paraphrases	Correct Ratio (%)	Model Accuracy (Avg.)	Model Accuracy (Std.)
all words from category	All	258	227	87.98%	.519	.074
	Rephrase	48	39	81.25%	.483	.035
	CoT	133	131	98.50%	.494	.065
	Gradual	74	54	72.97%	.604	.048
any words from category	All	259	233	89.96%	.443	.083
	Rephrase	48	44	91.67%	.451	.043
	CoT	135	134	99.26%	.438	.034
	Gradual	72	52	71.23%	.444	.160
ends with word	All	226	210	92.92%	.131	.024
	Rephrase	47	39	82.98%	.130	.022
	CoT	129	126	97.67%	.138	.019
	Gradual	47	42	89.36%	.112	.027
first alphabetically	All	233	198	84.98%	.326	.079
	Rephrase	47	38	80.85%	.293	.080
	CoT	121	117	96.69%	.315	.076
	Gradual	62	40	64.52%	.381	.053
homophones	All	264	234	88.64%	.252	.057
	Rephrase	48	43	89.58%	.214	.023
	CoT	140	128	91.43%	.246	.037
	Gradual	73	60	82.19%	.292	.081
less letters	All	240	207	86.25%	.338	.078
	Rephrase	42	40	95.24%	.316	.061
	CoT	126	119	94.44%	.319	.068
	Gradual	69	45	65.22%	.397	.074
more letters	All	237	210	88.61%	.374	.081
	Rephrase	45	42	93.33%	.349	.059
	CoT	127	123	96.85%	.359	.075
	Gradual	62	42	67.74%	.431	.078
rhyming word	All	245	219	89.39%	.234	.079
	Rephrase	47	37	78.72%	.189	.038
	CoT	125	115	92.00%	.198	.049
	Gradual	70	64	91.43%	.325	.067
word before	All	233	225	96.57%	.123	.049
	Rephrase	41	39	95.12%	.088	.009
	CoT	125	119	95.20%	.098	.012
	Gradual	64	64	100.0%	.195	.032
word not containing	All	234	223	95.30%	.222	.094
	Rephrase	48	47	97.92%	.177	.059
	CoT	125	122	97.60%	.190	.043
	Gradual	58	51	87.93%	.337	.115
all tasks	All	2429	2186	90.00%	.296	.070
	Rephrase	461	408	88.50%	.269	.043
	CoT	1286	1234	95.96%	.279	.048
	Gradual	652	514	78.83%	.351	.074

Table 7: The distribution of correct paraphrases for each generation method across all tasks in LMENTRY.

Table 25 contains the McNemar test p-values we used to assess the statistical significance of the differences in maximum performance between the original best prompt and the estimated optimal prompt.

Benchmark & Task	Method	#Auto Paraphrases	#Correct Paraphrases	Correct Ratio (%)	Model Accuracy (Avg.)	Model Accuracy (Std.)
causal judgement	All	187	153	81.82%	.477	.034
	Rephrase	50	31	62.00%	.469	.024
	CoT	60	55	91.67%	.452	.023
	Gradual	76	66	86.84%	.502	.028
disambiguation qa	All	188	177	94.15%	.412	.049
	Rephrase	50	50	100.0%	.403	.031
	CoT	60	50	83.33%	.357	.022
	Gradual	77	76	98.70%	.455	.029
formal fallacies	All	184	130	70.65%	.308	.026
	Rephrase	50	19	38.00%	.326	.027
	CoT	56	51	91.07%	.294	.015
	Gradual	77	59	76.62%	.313	.027
geometric shapes	All	178	171	96.07%	.163	.020
	Rephrase	50	50	100.0%	.175	.015
	CoT	55	53	96.36%	.153	.019
	Gradual	72	67	93.06%	.163	.020
hyperbaton	All	155	117	75.48%	.466	.035
	Rephrase	43	32	74.42%	.467	.020
	CoT	36	35	97.22%	.438	.030
	Gradual	75	49	65.33%	.484	.034
logical deduction five objects	All	189	150	79.37%	.262	.027
	Rephrase	50	47	94.00%	.239	.009
	CoT	59	27	45.76%	.243	.026
	Gradual	79	75	94.94%	.283	.015
logical deduction seven objects	All	186	145	77.96%	.236	.026
	Rephrase	50	41	82.00%	.215	.009
	CoT	60	31	51.67%	.219	.028
	Gradual	75	72	96.00%	.257	.016
logical deduction three objects	All	187	147	78.61%	.359	.044
	Rephrase	50	47	94.00%	.329	.023
	CoT	60	27	45.00%	.317	.030
	Gradual	76	72	94.74%	.394	.028
movie recommendation	All	180	164	91.11%	.348	.036
	Rephrase	47	47	100.0%	.371	.011
	CoT	57	50	87.72%	.323	.040
	Gradual	75	66	88.00%	.351	.032
navigate	All	170	152	89.41%	.386	.019
	Rephrase	50	50	100.0%	.374	.013
	CoT	54	54	100.0%	.388	.021
	Gradual	65	47	72.31%	.396	.017
penguins in a table	All	183	143	78.14%	.243	.026
	Rephrase	49	37	75.51%	.250	.014
	CoT	59	51	86.44%	.215	.007
	Gradual	74	54	72.97%	.265	.018
ruin names	All	157	143	91.08%	.254	.016
	Rephrase	50	49	98.00%	.252	.015
	CoT	40	35	87.50%	.250	.019
	Gradual	66	58	87.88%	.258	.013
salient translation error detection	All	136	128	94.12%	.191	.015
	Rephrase	47	46	97.87%	.185	.011
	CoT	25	25	100.0%	.192	.007
	Gradual	63	56	88.89%	.196	.019
snarks	All	162	152	93.83%	.405	.025
	Rephrase	50	50	100.0%	.396	.024
	CoT	37	37	100.0%	.410	.030
	Gradual	74	64	86.49%	.408	.021
sports understanding	All	173	137	79.19%	.461	.052
	Rephrase	48	31	64.58%	.469	.051
	CoT	57	49	85.96%	.453	.068
	Gradual	67	56	83.58%	.463	.035
all tasks	All	2615	2209	84.47%	.331	.035
	Rephrase	734	627	85.42%	.327	.025
	CoT	775	630	81.29%	.314	.030
	Gradual	1091	937	85.88%	.346	.029

Table 8: The distribution of correct paraphrases for each generation method across all tasks in BBH.

Task	Instruction Template #1	Instruction Template #2	τ
all words from category	Can you confirm if the list [{$words$}] consists solely of {$category$}? Please respond with either "yes" or "no".	Determine whether all the words in a given list belong to a specific category. The category is represented by the keyword {$category$}, and the list of words is represented by the keyword {$words$}. Answer the question with either 'yes' or 'no'. Category: {$category$} Words: {$words$} Output:	0.029
any words from category	Is {$category$} represented by any of these words {$words$}? Respond with either 'yes' or 'no'.	Given a list of words, determine if any of them belong to a specific category. Respond with either "yes" or "no". Words: {$words$} Category: {$category$} Output:	-0.200
ends with word	Provide a sentence that finishes with the term {$word$}.	Generate a sentence that ends with a specific word. Try to create a coherent sentence that effectively uses the provided word. Word: {$word$} Sentence:	-0.018
first alphabetically	Which word comes first alphabetically, "{$word1$}" or "{$word2$}"?	Please determine which of the two provided words is the first one alphabetically. The two words to be compared are denoted by placeholders {$word1$} and {$word2$}. Word 1: {$word1$} Word 2: {$word2$} Output: The first word alphabetically is	-0.095
homophones	Can you tell me which word, {$word1$} or {$word2$}, sounds like {$query$}?	Given two words, determine which one is a homophone or sounds more like a query word. Query word: {$query$} Word 1: {$word1$} Word 2: {$word2$} The word that sounds more like query is:	0.087
less letters	Which of {$word1$} and {$word2$} has fewer letters?	Compare two words and determine which one has fewer letters. The words are represented by the keywords {$word1$} and {$word2$}. Provide the keyword of the word with fewer letters. word1: {$word1$} word2: {$word2$} Output keyword:	-0.128
more letters	Please compare the length of "{$word1$}" and "{$word2$}" and provide the longer word.	Write a program that compares the length of two words and determines which one has more letters. Your program should take two words as input and output the word with more letters. Word 1: {$word1$} Word 2: {$word2$} Output:	-0.085
rhyming word	What is a word that rhymes with '{$query$}', '{$word1$}' or '{$word2$}'?	Given a query word and two candidate words, determine which candidate word rhymes with the query word. Your response should be the candidate word that rhymes with the query word. Query word: {$query$} Candidate word 1: {$word1$} Candidate word 2: {$word2$} Output word:	0.090
word before	Locate the word that comes immediately before '{$word$}' in the given sentence '{$sentence$}'	Given a sentence and a target word, identify the word that immediately precedes the target word in the sentence. Sentence: {$sentence$} Target word: {$word$} The word that comes right before {$word$} in the sentence is:	-0.099
word not containing	Create a term that does not have the inclusion of the letter "{$letter$}".	Write a word that does not contain the letter "{$letter$}". Letter: {$letter$} Output word:	-0.010

Table 9: Kendall τ values of the disagreement between ranks on models from example paraphrases for each task in LMENTRY.

Task	Instruction Template #1	Instruction Template #2	τ
causal judgement	You are required to give your opinion on the following question about causation: {$question$}. You must select either "yes" or "no".	Given a scenario, determine whether a typical person would attribute causality to a certain factor or not. Answer with "yes" or "no". Scenario: {$question$} Answer:	0.183
disambiguation qa	Q: For the given sentence, identify the antecedent of the ambiguous pronoun or state that it is ambiguous. Sentence: {$sentence$} Choose the option that correctly identifies the antecedent of the pronoun: {$options$} A:	Please clarify the meaning of the following sentence by selecting the option that correctly identifies the antecedent of the pronoun or state if it is ambiguous. Sentence: {$sentence$} Options: {$options$} Output:	0.164
formal fallacies	Q: "Classify the argument as either a formal fallacy or deductively valid. The explicitly stated premises are {$input$}." Options: - deductively valid - formal fallacy	Given a set of explicitly stated premises, determine whether the argument is deductively valid or a formal fallacy. Respond with "valid" or "invalid". Premises and conclusion: {$input$} Output:	-0.264
geometric shapes	Q: Identify the name of the geometric shape represented by the following SVG path element: {$svg\_path\_element$}. Options: {$options$} A:	From the given options {$options$}, select the name of the geometric shape that is represented by {$svg\_path\_element$}.	-0.267
hyperbaton	Order the adjectives correctly before a noun in English sentences, following the pattern of "[1. opinion] [2. size] [3. age] [4. shape] [5. color] [6. origin] [7. material] [8. purpose] noun". You will be presented with a multi-choice format question asking which sentence has the correct adjective order, with options provided. Which of the following sentences has the correct adjective order? {$options$}	Identify the sentence that has the correct order of adjectives in English. Choose the sentence that has the correct order of adjectives. Options: {$options$} Output:	0.019
logical deduction five objects	In this logical deduction task named logical deduction five objects, you will be given a set of paragraphs describing a set of five objects arranged in a fixed order. The statements are logically consistent within each paragraph. Your task is to choose the correct option from the given options. The options are {$options$}. The paragraph is {$paragraph$}.	Deduce the order of a sequence of five objects based on given logical statements. You will be given a set of logical statements and multiple choices for the order of the objects. Choose the correct order based on the given statements. Statements: {$paragraph$} Options: {$options$} Output:	0.264
logical deduction seven objects	Your task is to solve a logical deduction task which requires you to deduce the order of a sequence of objects. The task consists of a set of paragraphs, each describing a set of seven objects arranged in a fixed order. The statements are logically consistent within each paragraph. You will also be provided with multiple options to choose from. The options are represented by {$options$}. You should choose the correct option based on the information provided in the paragraph, which is represented by {$paragraph$}.	Deduce the order of a sequence of seven objects based on given statements. Use the provided options to answer each question. Statements: {$paragraph$} Options: {$options$} Output:	0.133
logical deduction three objects	Deduce the order of a sequence of three objects based on the logically consistent statements provided in the following {$paragraph$}. Choose the correct order from the given {$options$}.	Deduce the order of a sequence of three objects based on given statements and choose the correct option among multiple choices. Statements: {$paragraph$} Options: {$options$} Output:	0.056
movie recommendation	Q: Can you suggest a movie similar to {$movie\_list$}? Please choose from the following options: {$options$} A:	Based on a list of movies, recommend a similar movie from a set of options. Choose the option that best matches the given list. List of movies: {$movie\_list$} Options: {$options$} Output:	-0.018
navigate	Q: Would someone following {$instructions$} end up back at the starting point? Options: - Yes - No A:	Classify whether a series of navigation instructions will lead to the starting point or not. Provide either "yes" or "no" as the output. Instructions: {$instructions$} Output:	0.294
penguins in a table	Please answer the following question about the table of penguins: {$question$} The table has a header and each subsequent line represents a penguin with attributes: name, age, height (cm), weight (kg). {$table\_description$} You can choose from the following options: {$options$}	Given a table of penguins and their attributes, answer multiple choice questions about the penguins. The prompt will include a description of the table and several options to choose from. Table: {$table\_description$} Question prompt: {$question$} Options: {$options$} Output:	0.241
ruin names	Which of the following options is a funny way to "ruin" the name of {$artist\_or\_movie\_name$}? Choose from the following: {$options$}	Choose the option that best 'ruins' the {$artist\_or\_movie\_name$}: Options: {$options$} Answer:	0.075
salient translation error detection	Q: Identify the type of error in the given English translation of a German source sentence. The error will belong to one of the following categories: Named Entities, Numerical Values, Modifiers or Adjectives, Negation or Antonyms, Facts, or Dropped Content. The source sentence is {$source$}, and its translation is {$translation$}. From the options below, select the category that best describes the error. Options: {$options$}	Given an English translation of a German source sentence, identify the type of error present in the translation. The error can be one of the following types: Named Entities, Numerical Values, Modifiers or Adjectives, Negation or Antonyms, Facts, or Dropped Content. You must select the correct answer from the given options in a multi-choice format. German source sentence: {$source$} English translation: {$translation$} Options: {$options$}	0.389
snarks	Determine which of two sentences is sarcastic. The format is multi-choice and the prompt asks "which statement is sarcastic?" with two options provided. The correct option contains a taste of irony and satire. Options: {$options$} Output:	Given a question asking which statement is sarcastic and a list of options to choose from, determine which of the options is the sarcastic one. Question: "which statement is sarcastic?" Options: {$options$} Output:	0.22
sports understanding	Please classify the sentence "{$sentence$}" as either plausible or not plausible in the context of sports.	Is the following sentence plausible? {$sentence$} Answer with either "yes" or "no". Output:	-0.245

Table 10: Kendall τ values of the disagreement between ranks on models from example paraphrases for each task in BBH.

Task	Instruction Template #1	Instruction Template #2	τ
bbq lite	You are examined with your ability on social fairness by answering questions that contain bias content including: age, disability, gender, nationality, physical appearance, race ethnicity, religion, social status, and sexual orientation. $\{question\}$ Choose the most appropriate answer from: A) $\{choiceA\}$ B) $\{choiceB\}$ C) $\{choiceC\}$.	We are running a test for social biases. We will give you a question and a series of multiple-answers (A, B, C). Please choose from among the answers. Question: $\{question\}$ A: $\{choiceA\}$ B: $\{choiceB\}$ C: $\{choiceC\}$ Answer:	0.222
code line description	If you are an experienced code reviewer, please give an English language description of Python code$\{question\}$. Which is the most suitable? A. $\{choiceA\}$ B. $\{choiceB\}$ C. $\{choiceC\}$ D. $\{choiceD\}$ Answer:	You are given a simple line of Python code $\{question\}$. Try to find out its English equivalency from the following short sentences: A) $\{choiceA\}$, B) $\{choiceB\}$ C) $\{choiceC\}$, D) $\{choiceD\}$. The equivalent sentence is:	0.333
conceptual combinations	You are a linguistic expert that knows most of the concepts and combinations of words. Now, answer the following question: $\{context\}$ Question: $\{question\}$ (A) $\{choiceA\}$ (B) $\{choiceB\}$ (C) $\{choiceC\}$ (D) $\{choiceD\}$ Your answer is:	Question: $\{question\}$ The options are: A. $\{choiceA\}$ B. $\{choiceB\}$ C. $\{choiceC\}$ D. $\{choiceD\}$ Here is a context to help you answer the question: $\{context\}$. Choose the best answer from "A", "B", "C", "D".	0.182
hindu knowledge	In this task, you have to select the option that best answers the question given your knowledge about Hindu mythology. Question: $\{question\}$ A. $\{choiceA\}$ B. $\{choiceB\}$ C. $\{choiceC\}$ D. $\{choiceD\}$ Answer: among A, B, C, and D, the best choice is	$\{question\}$ A: $\{choiceA\}$ B: $\{choiceB\}$ C: $\{choiceC\}$ D: $\{choiceD\}$ With your expertise inhindu mythology, provide the correct answer:	0.444
known unknown	Verify if the question is unknown, choose your answer from options: Question: $\{question\}$ Options: A: $\{choiceA\}$ B: $\{choiceB\}$ Answer:	Question: $\{question\}$ To avoid hallucination, if the answer to this question is unknown, output "B", otherwise output "A"	-0.029
language identification	Please read the following sentence, then choose from the options which language you think it most likely came from. Your answer should be "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", or "K" Sentence: $\{question\}$ Options: A: $\{choiceA\}$ B: $\{choiceB\}$ C: $\{choiceC\}$ D: $\{choiceD\}$ E: $\{choiceE\}$ F: $\{choiceF\}$ G: $\{choiceG\}$ H: $\{choiceH\}$ I: $\{choiceI\}$ J: $\{choiceJ\}$ K: $\{choiceK\}$ Answer:	Please give the language used in the following sentence. Each sentence will give five options, please output the corresponding option (i.e. A, B, C, D, E, F, G, H, I, J, or K) to represent the corresponding answer. Sentence: $\{question\}$ Options:	0.028
logic grid puzzle	You are given a logic grid puzzle to test your sense of space and positions. You are given a context and some clues to pick the correct answer from the options to answer a question.Context: $\{context\}$ $\{clues\}$ Question: $\{question\}$ Options: (A) $\{choiceA\}$ (B) $\{choiceB\}$ (C) $\{choiceC\}$ (D) $\{choiceD\}$ (E) $\{choiceE\}$ Answer:	You are a master at solving logic grid puzzles. Solve this: $\{context\}$ $\{clues\}$ $\{question\}$	0.327
logical deduction	Given the following text describing the correct order of five objects, select the option from (A, B, C, D or E) that is consistent with the text. text: $\{question\}\{options\}$ answer:	The following text describes the arrangement order of five objects. Please read the text and choose the one from the options that matches the logic of the text description. Your answer should be "A", "B", "C", "D" or "E". Text: $\{question\}\{options\}$ Answer:	0.667
novel concepts	You are given three objects $\{question\}$, choose the option from below where the objects share the greatest similarity. A. $\{choiceA\}$ B. $\{choiceB\}$ C. $\{choiceC\}$ D. $\{choiceD\}$ E. $\{choiceE\}$	$\{question\}$ Pick the most correct description from: A.$\{choiceA\}$ B. $\{choiceB\}$ C. $\{choiceC\}$ D. $\{choiceD\}$ E. $\{choiceE\}$ My answer is:	0.400
play dialog	Now you are a dramatist. The following transcripts of dialogues are taken from Shakespeare plays, but the transcripts do not mark who said what. Your task is to identify whether the sentences in question were spoken by the same or different people. Here is the play: $\{play\}$ Question: In the preceding dialogue, were the lines $\{line1\}$ and $\{line2\}$ spoken by the same person or different people? Please just give a short answer: same or different. Your Answer:	In the context of the Shakespeare play, $\{play\}$, assess the given dialogue transcripts. Determine whether the sentences $\{line1\}$ and $\{line2\}$ were spoken by a single person or by different people. Answer:	-0.638
strange stories	Given a story, answer whether the question is true or false. $\{context\}$ Q: $\{question\}$ A:	Image you are taking a psychology test. Please read the given story and answer the question. Please answer "yes" or "no". Story: $\{context\}$ Q: $\{question\}$ A:	0.310
strategic qa	Reason about the answer to the question. $\{question\}$	Please answer the following question, you should think step by step, but please use "yes" or "no" to answer.Question: $\{question\}$Answer:	-0.085
vitaminc fact verification	Input: $\{claim\}$ Verify the factually of the claim based on the following context $\{context\}$ - "True" if the claim is factually correct - "False" if the claim is factually incorrect - "Neither" if the factuality cannot be determined. Output you answer with one of "True", "False", or "Neither". Answer:	Context: $\{context\}$ Now classify this claim into one of 'True', 'False', or 'Neither'. $\{claim\}$	0.556
winowhy	Read the following reasoning about who a particular pronoun refers to: $\{question\}$ Is the reasoning correct?	Read the following reasoning, and answer if its correct or incorrect. $\{question\}$	-0.056

Table 11: Kendall τ values of the disagreement between ranks on models from example paraphrases for each task in BBL.

		Count	t03b	t0pp	fal7b	mpt7b	alp7b	alp13b	ft5-s	ft5-b	ft5-l	ft5-xl	ft5-xxl	air13b	nou13b	ult13b	vic13b	min15b
Add char	[]->.	11	-0.01	-0.07	0.13	0.02	0.00	0.16	-0.03	0.00	-0.02	-0.13	0.01	-0.01	-0.02	0.06	0.01	0.01
Change char	. -> :	18	0.00	0.05	0.01	0.01	0.06	-0.19	0.01	-0.02	0.02	0.00	-0.01	0.00	0.07	0.00	-0.01	0.00
Add 2 chars	… -> "…"	43	0.03	0.01	0.00	0.01	0.00	0.04	-0.11	-0.05	0.06	-0.04	0.06	0.03	0.00	0.04	0.01	0.00
	… -> '…'	7	0.03	0.04	-0.01	0.04	-0.01	0.14	-0.06	0.00	0.10	-0.01	0.04	0.02	0.03	0.02	0.05	0.00
Add word	Q:	5	-0.09	-0.30	0.00	-0.56	-0.01	0.00	0.00	0.04	0.00	-0.01	0.02	-0.34	-0.20	-0.29	-0.01	-0.61
	these	5	0.09	-0.02	0.18	-0.08	0.01	0.02	0.12	-0.03	0.10	-0.05	0.06	0.09	0.09	-0.02	-0.02	-0.03
	more	15	0.04	0.00	0.09	0.02	0.02	0.00	0.04	0.09	0.08	0.01	-0.02	0.14	-0.07	-0.07	-0.01	-0.05
	using	6	0.00	-0.06	0.42	0.01	0.01	0.13	-0.02	-0.01	-0.25	0.01	0.02	0.02	0.10	-0.03	0.07	0.01
Change word	create -> write	14	0.00	-0.02	0.14	-0.01	0.04	0.02	-0.03	0.02	-0.02	0.05	0.00	0.01	-0.08	0.13	-0.10	0.02
	sentence -> statement	9	0.01	0.03	-0.01	0.02	0.02	-0.05	-0.09	0.00	0.02	0.00	0.00	-0.15	-0.01	-0.01	-0.05	-0.02
	create -> generate	12	0.01	0.06	-0.01	-0.03	0.00	0.13	-0.02	-0.02	-0.03	0.07	-0.01	-0.02	0.02	-0.02	0.06	0.01
	compose -> write	18	0.00	0.02	0.27	-0.02	0.02	0.05	-0.03	0.00	0.01	0.18	0.00	-0.01	0.05	0.18	-0.01	0.01
	produce -> write	7	-0.01	0.01	0.07	0.00	-0.01	-0.02	0.00	0.01	0.00	0.05	-0.04	-0.02	0.06	0.14	-0.12	0.02
	appears -> comes	21	-0.03	0.00	0.22	0.14	0.02	0.03	0.01	-0.04	-0.01	-0.04	-0.03	0.11	0.06	0.15	0.12	0.17
	and -> or	23	0.01	0.02	0.00	0.01	0.00	-0.02	0.23	0.13	0.10	0.03	-0.03	-0.02	0.03	0.04	-0.02	0.02
	a -> the	9	0.00	0.01	0.02	0.15	0.01	0.02	0.03	0.00	0.03	-0.03	0.04	-0.02	0.00	0.04	0.02	0.00
	lesser -> smaller	6	0.01	0.01	-0.01	0.17	-0.01	-0.02	-0.01	-0.02	-0.04	0.00	-0.02	0.02	0.08	-0.02	0.00	-0.02
	generate -> provide	6	-0.12	-0.06	0.01	0.01	0.01	-0.11	0.01	0.02	0.02	0.14	0.02	0.02	-0.01	-0.02	0.00	-0.08
	excludes -> omits	6	0.00	0.10	-0.08	0.00	0.00	0.04	0.13	-0.01	-0.01	0.16	0.01	0.03	0.13	0.21	-0.02	-0.01
	excludes -> lacks	11	0.00	0.07	0.15	0.00	0.00	0.09	0.12	-0.04	-0.28	0.46	-0.01	0.02	-0.07	0.06	-0.07	-0.01
	lacks -> omits	5	0.00	0.05	-0.18	0.00	0.00	-0.11	0.03	0.00	0.04	-0.22	0.02	-0.02	0.27	0.32	0.07	-0.02
	contain -> have	9	-0.02	-0.02	-0.32	0.01	0.00	0.00	0.03	-0.03	-0.01	0.11	0.06	-0.05	0.10	-0.01	0.01	-0.01
	have -> include	5	-0.09	-0.11	0.45	-0.01	0.00	-0.04	-0.02	0.00	0.13	-0.11	-0.05	0.02	0.05	0.25	-0.01	0.00

Figure 8: Average performance differences between various models when small edits are made to the prompts (e.g., substituting 'excludes' with 'lacks'). The count column describes the number of tasks for which this edit was relevant.

Change	Model	P1	Acc.	P2	Acc.	Diff.
'.' –>':'	nous-hermes	Create a word that does not include the letter "{letter}".	.04	Create a word that does not include the letter "{letter}"***:***	.65	+.62
	alpaca-13b	Create a sentence that concludes with the term "{word}".	.61	Create a sentence that concludes with the term "{word}"***:***	.19	-.42
+ '.'	alpaca-13b	Write a word that lacks the letter "{letter}"	.04	Write a word that lacks the letter "{letter}"**.**	.42	+.38
	falcon-7b	Write a word that lacks the letter "{letter}"	.19	Write a word that lacks the letter "{letter}"**.**	.50	+.31
	flan-t5-xl	Write a word that omits the letter "{letter}"	.77	Write a word that omits the letter "{letter}"**.**	.54	-.23
+ "..."	mpt-7b	Write a word that does not contain the letter ***{letter}***.\nWord:	.58	Write a word that does not contain the letter ***"{letter}"***.\nWord:	.19	-.38
	flan-t5-small	Write a word that does not contain the letter ***{letter}***.\nWord:	.62	Write a word that does not contain the letter ***"{letter}"***.\nWord:	.04	-.58
	flan-t5-xl	Write a word that excludes the letter ***{letter}***.	.81	Write a word that excludes the letter ***"{letter}"***.	.23	-.58
+ 'Q:'	minotaur	Are all of the words in the set {words} classified as {category}? Please respond with either 'yes' or 'no'.	.69	***Q:*** Are all of the words in the set {words} classified as {category}? Please respond with either 'yes' or 'no'.	.02	-.67
	airoboros	Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.75	***Q:*** Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.09	-.66
	mpt-7b	Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.57	***Q:*** Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.00	-.57
	t0pp	Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.99	***Q:*** Are all the words in {words} categorized as {category}? Please answer with either 'yes' or 'no'.	.55	-.44
+ 'using'	flan-t5-large	Your task is to write a word without the letter "{letter}".	.46	Your task is to write a word without ***using*** the letter "{letter}".	.12	-.35
	falcon-7b	Write a word without the letter {letter}.\nOutput word:	.12	Write a word without ***using*** the letter {letter}.\nOutput word:	.35	+.23
	flan-t5-large	Write a word without the letter {letter}.\nOutput word:	.73	Write a word without ***using*** the letter {letter}.\nOutput word:	.50	-.23
omits –>lacks	ultralm-13b	Write a word that ***omits*** the letter "{letter}".	.62	Write a word that ***lacks*** the letter "{letter}".	.19	-.42
	falcon-7b	Write a word that ***omits*** the letter "{letter}".	.19	Write a word that ***lacks*** the letter "{letter}".	.50	+.31
	flan-t5-xl	Write a word that ***omits*** the letter "{letter}".	.54	Write a word that ***lacks*** the letter "{letter}".	.81	+.27
contain –>have	falcon-7b	Write a word that does not ***contain*** the letter "{letter}".	.81	Write a word that does not ***have*** the letter "{letter}".	.19	-.62
	falcon-7b	Write a word that does not ***contain*** the letter "{letter}".	.81	Write a word that does not ***have*** the letter "{letter}".	.27	-.54
	flan-t5-xxl	Please write a word that does not ***contain*** the letter "{letter}".	.62	Please write a word that does not ***have*** the letter "{letter}".	.88	+.27
include –>have	falcon-7b	Write a word that does not ***include*** the letter "{letter}".	.81	Write a word that does not ***have*** the letter "{letter}".	.19	-.62
	flan-t5-xl	Write a word that does not ***include*** the letter "{letter}".	.42	Write a word that does not ***have*** the letter "{letter}".	.73	+.31
	falcon-7b	Please write a word that does not ***include*** the letter "{letter}".	.77	Please write a word that does not ***have*** the letter "{letter}".	.35	-.42
	ultralm-13b	Please write a word that does not ***include*** the letter "{letter}".	.46	Please write a word that does not ***have*** the letter "{letter}".	.12	-.35
excludes –>lacks	flan-t5-large	Write a word that ***excludes*** the letter "{letter}".	.54	Write a word that ***lacks*** the letter "{letter}".	.12	-.42
	flan-t5-xl	Write a word that ***excludes*** the letter "{letter}".	.19	Write a word that ***lacks*** the letter "{letter}".	.81	+.62
	flan-t5-xl	Write a word that ***excludes*** the letter "{letter}"	.46	Write a word that ***lacks*** the letter "{letter}"	.88	+.42

Table 12: Representative examples of instruction template pairs from LMENTRY with very minor differences but notable variations in performance (open-source models).

Benchmark & Task	Kendall's W	Friedman p-val
BIG-bench Lite		
known unknown	.316	**4.4E-5**
play dialog	.355	**4.3E-5**
winowhy	.520	**6.0E-4**
strategic qa	.529	.191
hindu knowledge	.560	.569
conceptual combinations	.731	.132
strange stories	.731	.431
code line description	.756	**.002**
novel concepts	.787	.620
logic grid puzzle	.796	**.010**
language identification	.811	**.002**
vitaminc fact verification	.888	.772
bbq lite	.890	**.023**
logical deduction	.913	.895

Table 13: Kendall's $W \in [0, 1]$ values for all tasks sorted in ascending order. The smaller the value of W the more that the ranking on different prompts is decorrelated. Most W are smaller than 0.85, indicating less than optimal correlation. The p-values from the Friedman test indicate significant differences between rankings of models when using different prompts for 7 tasks.

BBH tasks

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
t0pp	0.60	0.25	-0.74	-0.71	0.23	-1.13	-1.03	0.40	1.70	1.83	1.54	-1.09	0.79	0.05	0.08
falcon-7b-instruct	-2.39	-0.36	-0.85	0.31	0.05	-0.01	-0.49	-1.03	0.03	0.23	-0.11	0.28	-2.12	-0.04	-0.59
mpt-7b-instruct	0.77	0.59	-0.08	-0.04	0.03	1.97	1.39	0.27	-0.63	-0.40	1.32	-0.18	0.64	1.38	-0.50
alpaca-13b	-0.16	-0.37	0.12	0.81	-0.39	1.11	-0.95	-0.20	-1.97	-0.13	-0.59	-1.75	4.03	-0.30	-0.86
flan-t5-xl	-1.04	-1.36	-0.55	0.93	1.32	-0.76	-0.87	-0.66	1.59	0.35	-0.31	0.19	-2.27	1.10	0.10
flan-t5-xxl	-0.96	-0.57	0.24	-0.29	-0.05	-0.91	-0.47	-0.73	-0.04	0.61	-0.49	0.15	-0.23	-0.04	-0.63
airoboros-13b	-1.43	-0.28	-0.20	0.43	0.39	0.05	0.17	0.90	0.69	-1.99	1.97	0.75	-0.47	-1.24	-0.72
nous-hermes-13b	-1.32	0.44	-0.17	-0.89	0.05	-0.42	-0.78	-0.20	-0.82	0.19	0.80	1.38	-1.26	0.31	0.43
ultralm-13b	-0.65	0.55	-0.27	-0.41	1.11	0.56	0.53	0.71	1.03	0.46	-0.15	-0.02	-0.50	-0.49	0.47
vicuna-13b	0.08	-0.72	-0.93	0.33	-0.55	0.77	0.27	0.28	1.58	0.48	-0.51	2.55	0.35	0.81	-0.28
minotaur-15b	-0.08	-0.55	-0.09	-0.70	-0.34	-0.74	-0.81	-0.79	-0.77	-0.53	0.60	-0.63	-0.30	-0.74	-0.52

Figure 9: Model and task performance divergence. For each task, this table shows the number of standard deviations by which the performance of each model on the original prompts deviates from the average model performance. Dark red cells indicate substantial divergence values exceeding one standard deviation.

BBL tasks

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
t0_3b	1.12	1.22		-0.28	-0.86	2.39		0.68	-1.01	0.83	0.58	-1.16	0.89	-0.87
t0pp	-0.47	-0.38	1.60	-2.11	-0.20	0.80	0.66	0.23	0.55	0.17	0.22	-1.54	0.45	-1.67
alpaca-7b	0.30	-2.38	-0.02	0.26	-0.98	0.75	-1.58	-2.15	0.19	0.10	-0.67	0.57	-0.87	-0.32
alpaca-13b	2.47	-2.51	-0.16	0.75	-1.29	1.11	0.28	-0.24	-0.21	1.44	-0.70	0.94	-0.71	-0.79
flan-t5-small	0.04	0.72	1.19	-0.10	1.28	-0.26	0.21	-0.05	-0.16	0.33	-0.58	0.75	0.02	-1.85
flan-t5-base	-0.50	-2.07	-0.03	2.31	-1.58	0.27	0.04	-2.21	0.49	0.63	-1.13	-2.46	0.38	-1.60
flan-t5-large	-0.14	-2.62	-1.37	-0.17	-1.67	-0.41	1.31	-2.65	-1.85	0.74	-2.37	-0.49	-0.23	-1.38
flan-t5-xl	-0.75	-2.82	-1.17	-1.15	-1.28	0.03	0.49	0.90	-0.62	0.61	2.61	-1.26	0.51	-2.31
flan-t5-xxl	0.14	-0.75	-1.04	-1.71	-0.82	-0.42	-0.64	1.65	0.16	1.08	0.93	-1.00	1.09	-1.42

Figure 10: Model and task performance divergence. For each task, this table shows the number of standard deviations by which the performance of each model on the original prompts deviates from the average model performance. Dark red cells indicate substantial divergence values exceeding one standard deviation.

Benchmark & Task	MaxP	AvgP	Sat	Combined
LMENTRY				
all words from category	.958	.967	.950	.900
any words from category	.979	.967	.983	.983
ends with word	.104	.967	.050	.517
first alphabetically	1.00	.950	.933	.950
homophones	.945	1.00	.900	.933
less letters	.983	1.00	.917	.967
more letters	.970	.983	.917	.983
rhyming word	1.00	.967	.983	.967
word before	1.00	1.00	.983	1.00
word not containing	.836	.967	.783	.850
BIG-bench Hard				
causal judgement	1.00	1.00	1.00	1.00
disambiguation qa	1.00	1.00	.964	1.00
formal fallacies	.991	.927	.818	.855
geometric shapes	1.00	.964	1.00	.964
hyperbaton	.953	1.00	.927	.964
logical deduction five objects	1.00	.964	.855	.964
logical deduction seven objects	1.00	1.00	.964	.964
logical deduction three objects	1.00	.964	.891	.964
movie recommendation	.954	.927	.891	.964
navigate	.964	1.00	1.00	.964
penguins in a table	1.00	1.00	.891	1.00
ruin names	1.00	1.00	.964	.891
salient translation error detection	1.00	1.00	1.00	1.00
snarks	1.00	1.00	.964	1.00
sports understanding	1.00	1.00	.964	1.00

Table 14: Kendall's Tau model ranking comparisons before and after removal of incorrect paraphrases. Results show near-perfect to perfect agreement across all tasks, except for LMENTRY's "ends with word" task.

	average	maximum	saturation	combined
t0_3b	7.40	12.20	5.90	7.90
t0++	4.80	8.80	4.20	4.63
falcon-7b	9.40	8.00	9.70	8.27
mpt-7b	10.30	11.10	10.00	10.00
alpaca-7b	13.90	7.90	13.60	13.20
alpaca-13b	11.50	6.60	12.40	10.72
flan-t5-small	6.20	9.60	7.90	5.90
flan-t5-base	8.00	10.50	6.50	8.20
flan-t5-large	4.10	8.50	4.80	4.50
flan-t5-xl	2.80	4.20	5.80	3.72
flan-t5-xxl	**1.40**	**3.10**	**3.70**	**1.72**
airoboros-13b	9.50	10.20	9.50	10.36
nous-hermes-13b	8.90	6.30	9.70	8.73
ultralm-13b	12.30	9.80	9.30	11.09
vicuna-13b	11.80	3.70	14.70	10.63
minotaur-15b	13.70	10.20	8.30	13.63

Table 15: Average model ranks for each metric across all tasks in LMENTRY. Bold numbers indicate the best averaged rank per metric, while underlined numbers indicate the worst averaged rank per metric.

	average	maximum	saturation	combined
t0++	7.33	7.47	5.67	7.33
falcon-7b	6.20	7.60	4.67	6.47
mpt-7b	9.00	9.33	8.00	9.53
alpaca-13b	4.53	5.27	3.93	4.67
flan-t5-xl	2.67	2.47	3.93	2.67
flan-t5-xxl	**1.40**	**1.87**	**3.20**	**1.33**
airoboros-13b	5.73	5.67	5.80	5.67
nous-hermes-13b	6.87	5.40	8.13	6.67
ultralm-13b	8.53	6.60	8.73	8.20
vicuna-13b	3.07	3.13	5.27	3.13
minotaur-15b	10.67	9.67	8.67	10.33

Table 16: Average model ranks for each metric across all tasks in BBH. Bold numbers indicate the best averaged rank per metric, while underlined numbers indicate the worst averaged rank per metric.

model	default	rephrase	cot	gradual
t0_3b (*)	0.00	11.76	**58.82**	29.41
t0++ (*)	0.00	15.00	**45.00**	40.00
falcon-7b	9.09	9.09	36.36	**45.45**
mpt-7b	0.00	23.53	**47.06**	29.41
alpaca-7b (**)	8.33	0.00	0.00	**91.67**
alpaca-13b (**)	0.00	0.00	8.33	**91.67**
flan-t5-base (*)	0.00	7.14	**64.29**	28.57
flan-t5-small (*)	0.00	0.00	**58.33**	41.67
flan-t5-large (*)	0.00	**40.00**	**40.00**	20.00
flan-t5-xl (*)	0.00	7.69	30.77	**61.54**
flan-t5-xxl (*)	0.00	13.04	**69.57**	17.39
airoboros-13b (**)	0.00	35.71	14.29	**50.00**
nous-hermes-13b (**)	0.00	0.00	33.33	**66.67**
ultralm-13b (**)	0.00	6.67	**66.67**	26.67
vicuna-13b (**)	10.00	10.00	0.00	**80.00**
minotaur-15b	0.00	14.29	28.57	**57.14**
all models	**1.33**	**12.39**	**40.71**	**45.58**
all paraphrases	**1.24**	**18.98**	**52.94**	**26.84**

Table 17: Distribution of optimal paraphrase sources per model for LMENTRY. Rows represent models, with T5-based models marked by an asterisk (*) and LLaMA-based models by two asterisks (**). Columns indicate paraphrase generation methods. Percentages in each cell show the rate of optimal paraphrases from each method, with bold numbers identifying the leading source for each model. The 'All Models' row aggregates percentages across all models, while the 'All Paraphrases' row displays the overall distribution of generation methods across all paraphrases.

model	default	rephrase	cot	gradual
falcon-7b	0.00	27.27	18.18	**54.55**
mpt-7b	0.00	40.00	16.00	**44.00**
flan-t5-xl	2.38	23.81	26.19	**47.62**
flan-t5-xxl	4.35	17.39	26.09	**52.17**
t0++	0.00	**50.00**	40.00	10.00
alpaca-13b	0.00	11.11	**44.44**	**44.44**
airoboros-13b	0.00	**50.00**	16.67	33.33
nous-hermes-13b	0.00	12.50	16.67	**70.83**
ultralm-13b	0.00	29.17	25.00	**45.83**
vicuna-13b	0.00	**38.10**	28.57	33.33
minotaur-15b	0.00	9.09	0.00	**90.91**
all tasks	**0.73**	**27.37**	**23.72**	**48.18**
all paraphrases	**0.57**	**28.07**	**29.64**	**41.72**

Table 18: Distribution of optimal paraphrase sources per model for BBH. Rows represent models and columns indicate paraphrase generation methods. Percentages in each cell show the rate of optimal paraphrases from each method, with bold numbers identifying the leading source for each model. The 'All Models' row aggregates percentages across all models, while the 'All Paraphrases' row displays the overall distribution of generation methods across all paraphrases.

task	t0_3b	t0++	fal7b	mpt7b	alp7b	alp13b	ft5small	ft5base	ft5large	ft5xl	ft5xxl	airoboros	noushermes	ultralm	vicuna	minotaur
all words from category	4.9	2.2	11.3	4.2	3.1	2.9	5.6	4.3	2.4	6.6	3.1	6.6	8.7	3.2	3.8	7.5
any words from category	3.4	1.5	11.2	4.1	3.8	2.9	11	7.3	6.7	2	1.7	5.8	3.2	6	2.6	2.1
ends with word	3.3	4.3	2.2	2.4	1.9	10.7	3.3	6.2	5.7	5.9	6.1	2.8	3.5	2	3.7	2.5
first alphabetically	10.3	3.2	7.2	6.1	2.6	6.4	10.3	3.2	5.1	3.9	4	5.3	8.2	11.4	5.7	7.8
homophones	7.2	5.7	8.3	10.3	2.4	4.2	10.5	4.5	3.5	3.7	9.7	5.8	10.8	2.1	3.7	1.2
less letters	3.3	9.5	5.3	3.8	4.2	5.5	5.9	4.6	4.5	5.1	3.5	4.2	3.5	7.1	10.2	6.3
more letters	4.3	4.2	6.5	6	5.8	10.4	3.3	4.2	4.2	4.9	7.6	5.4	9.3	6	7.9	6.8
rhyming word	10.9	2.5	4.1	5.2	2.6	12.8	6.6	3.8	6.7	5.1	5.6	3.3	3.8	1.8	7.3	1.2
word before	5.9	2.8	1.3	3.2	6.6	3.5	6	4.4	4.5	8.8	4.9	4.5	3.3	1.1	5.1	2
word not containing	4.2	12.4	8.6	11.8	9.4	6.3	4.4	10	21.9	14.2	5.8	10.1	5.8	6.3	5	12.3

Table 19: The average number of average heuristic repetitions required to achieve less than a 1 accuracy point discrepancy from the actual average performance for each task and open-source model. Maximal value: 21.9. All values average: 5.62 (std: 3.12).

task	t0_3b	t0++	fal7b	mpt7b	alp7b	alp13b	ft5small	ft5base	ft5large	ft5xl	ft5xxl	airoboros	noushermes	ultralm	vicuna	minotaur
all words from category	✓	✓	0.01	✓	✓	✓	0.03	✓	0.02	✓	✓	✓	0.02	✓	✓	✓
any words from category	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.06	✓	✓	✓
ends with word	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.03
first alphabetically	✓	✓	✓	✓	✓	✓	0.01	✓	0.03	0.03	✓	✓	✓	✓	✓	✓
homophones	✓	✓	✓	✓	✓	✓	✓	✓	0.04	✓	✓	✓	✓	✓	✓	0.01
less letters	✓	0.02	0.01	✓	✓	✓	0.02	✓	0.03	✓	0.02	✓	✓	✓	✓	✓
more letters	0.01	0.01	✓	0.03	✓	✓	✓	✓	✓	0.01	✓	✓	✓	0.01	✓	✓
rhyming word	✓	0.01	✓	✓	✓	✓	0.02	0.03	0.01	0.01	✓	✓	✓	0.01	✓	✓
word before	✓	✓	✓	✓	0.01	✓	✓	✓	✓	✓	0.06	✓	✓	✓	0.01	✓
word not containing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 20: Results of the greedy optimal paraphrase search for each task and open-source model. An optimal prompt was recovered in 130 out of 160 cases. In the remaining cases, the average discrepancy in performance between the chosen and actual optimal paraphrases was 2.1 accuracy points, with a standard deviation of 1.4.

	davinci	td002	td003	cgpt
all words from category	0.56	0.72	0.84	0.60
any words from category	0.55	0.63	0.65	0.86
ends with word	0.10	0.30	0.58	0.60
first alphabetically	0.46	0.48	0.71	0.98
homophones	0.48	0.19	0.38	0.49
less letters	0.41	0.67	0.79	0.88
more letters	0.47	0.68	0.82	0.87
rhyming word	0.19	0.29	0.57	0.69
word before	0.12	0.13	0.27	0.40
word not containing	0.03	0.85	0.97	0.90

Table 21: Average performances for OpenAI models across all LMENTRY tasks, computed using only the original prompts.

	davinci	td002	td003	cgpt
all words from category	0.64	0.80	0.85	0.68
any words from category	0.66	0.82	0.88	0.97
ends with word	0.15	0.35	0.61	0.62
first alphabetically	0.50	0.56	0.90	0.98
homophones	0.59	0.25	0.41	0.79
less letters	0.48	0.70	0.86	0.92
more letters	0.54	0.80	0.89	0.90
rhyming word	0.32	0.45	0.65	0.96
word before	0.17	0.21	0.34	0.66
word not containing	0.04	0.92	1.00	0.96

Table 23: Max performances for OpenAI models across all LMENTRY tasks, computed using only the original prompts.

	davinci	td002	td003	cgpt
all words from category	0.15	0.61	0.79	0.62
any words from category	0.16	0.62	0.59	0.82
ends with word	0.11	0.24	0.42	0.54
first alphabetically	0.12	0.27	0.35	0.45
homophones	0.12	0.57	0.60	0.71
less letters	0.17	0.51	0.58	0.58
more letters	0.16	0.49	0.51	0.50
rhyming word	0.14	0.39	0.41	0.76
word before	0.04	0.16	0.51	0.47
word not containing	0.06	0.57	0.84	0.81

Table 22: Estimated average performances for OpenAI models across all LMENTRY tasks, approximated using all prompt paraphrases.

	davinci	td002	td003	cgpt
all words from category	0.64	0.94	0.99	0.97
any words from category	0.66	0.95	0.99	1.00
ends with word	0.88	0.52	0.67	0.72
first alphabetically	0.55	0.95	0.97	1.00
homophones	0.63	0.99	0.95	0.99
less letters	0.61	0.95	0.95	1.00
more letters	0.71	0.93	0.97	1.00
rhyming word	0.67	0.93	0.95	0.99
word before	0.26	0.51	0.82	0.95
word not containing	0.65	1.00	1.00	1.00

Table 24: Estimated max performances for OpenAI models across all LMENTRY tasks, approximated using all prompt paraphrases.

task	davinci	td002	td003	cgpt
all words from category	1	.0009	.0002	7.2E-08
any words from category	1	.0008	.0009	.0832
ends with word	8.8E-17	.0195	.3034	.0955
first alphabetically	.4922	6.1E-09	.0196	.1572
homophones	.5371	7.8E-18	5.3E-13	7.7E-06
less letters	.0633	3.4E-06	.0009	.0047
more letters	.0131	.0046	.0209	.0016
rhyming word	5.7E-07	1.4E-10	4.3E-08	.0833
word before	.10560	1.1E-06	1.1E-11	1.9E-06
word not containing	6.3E-05	.1573	1	.3173

Table 25: The results for the McNemar test we ran to assess the statistical significance of the differences in maximum performance between the original best prompt and the prompt estimated to be optimal across all paraphrases for each task in the LMENTRY benchmark. Significant max differences (p-value<0.05) are highlighted.