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, Israel
[◇]Department of Information Systems, University of Haifa, Israel

{moran.mizrahi, guy.kaplan2, dan.malkinhueb,
gabriel.stanovsky}@mail.huji.ac.il
rdror@is.haifa.ac.il, dshahaf@cs.huji.ac.il

Abstract

Recent advances in LLMs have led to an abundance of evaluation benchmarks, which typically rely on a *single instruction template* per task. We create a large-scale collection of instruction paraphrases and comprehensively analyze the brittleness introduced by 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 performance, both absolute and relative. 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., 2023; Liang et al., 2023). 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.

We explore the question of *robustly comparing different models on a given task*. We first create a dataset of paraphrased instructions, employing three automatic paraphrasing methods based on

recent 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]

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. For example, Figure 1 shows four models evaluated on four semantically equivalent prompts, with both absolute and relative performance varying widely; one can even observe cases where the same model performs *the best* on one instruction and *the worst* on a semantically equivalent instruction (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 single-instruction evaluation. This may also partially explain why some models seem less accurate in practice than their formal evaluation suggests.

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-prompt LLM evaluation* — a set of metrics which measure aggregated performance over a set of instruction template paraphrases.

We argue that different use cases should entail different evaluation metrics. For example, LLM developers may be interested in measuring

---

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

933

LMentry: Homophones
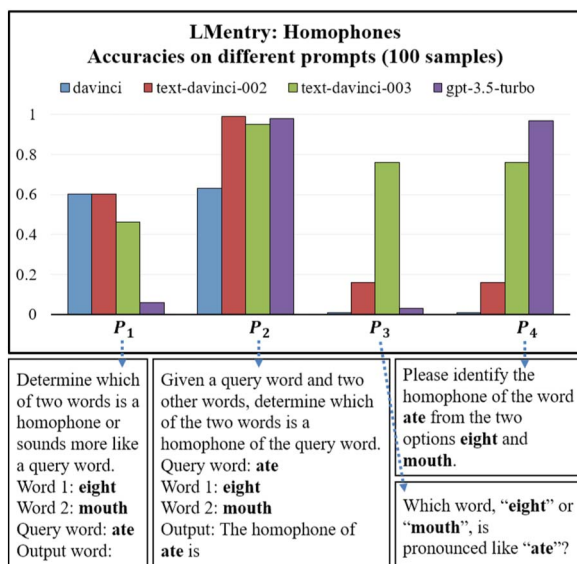**Accuracies on different prompts (100 samples)**

Figure 1: Evaluation of different OpenAI models on the homophones task from LMENTRY over four paraphrases. Each cluster of columns corresponds to a distinct paraphrased *instruction template* (see respective texts below; words in bold indicate an instantiation). Despite all instructions being semantically equivalent, both absolute performance and relative ranking vary widely.

the *robustness of performance* across multiple instruction templates. In contrast, developers aiming to integrate an LLM into a specific downstream task may be interested in comparing models according to their corresponding top-performing instruction.

We evaluate 20 LLMs with our metrics, finding that their absolute and relative performance differ from results obtained with the benchmarks' original instructions. 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, but 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 use multi-prompt LLM evaluations and choose a metric for aggregating the results according to 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 the common practice involves a single (or very few) task instruction templates. In the rest of the paper, we will argue that such practice leads to brittle, unreliable results.

Task Instruction Templates. Following Mishra et al. (2022) and Chung et al. (2024), 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., 2023; Suzgun et al., 2023), and HELM (Liang et al., 2023). In all of these, each task has a single instruction template, against which all models are evaluated. Another benchmark, LMENTRY (Efrat et al., 2023), 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.

We note that many notable works do not disclose the instruction templates used for evaluation (e.g., LLaMA [Touvron et al., 2023], PALM [Chowdhery et al., 2023], GPT-4 [Achiam et al., 2023], Gemini [Team et al., 2023]). While there are reasons to withhold instructions (e.g., avoid potential leakage), this practice exacerbates the challenge of meaningful comparative evaluation.

Prompt Robustness. Related to this study is a line of work measuring LLMs' 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. (2023) examined a

934

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 compared to manual paraphrases. We later incorporate their manual paraphrases in our evaluation of BIG-bench Lite.

In contrast to works on prompt robustness, 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

3.1 Tasks

We evaluate 39 diverse tasks from three evaluation benchmarks, as itemized below.

10 Tasks from LMENTRY (Efrat et al., 2023). 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 the 10 tasks that received the lowest scores in the original paper, as these more challenging tasks are likely to better highlight the differences between models.

14 Tasks from BIG-bench Lite (BBL; Srivastava et al., 2023). These cover multiple knowledge domains, sampled from the larger BIG-Bench benchmark (Srivastava et al., 2023). 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., 2023). This is another curated subset of BIG-bench, containing particularly challenging tasks on which LLMs underperform the average human score. We focused on a set of 15 classification and multiple choice tasks to streamline the evaluation process. Each task in BBH is associated with a single instruction template.

Measuring Performance. In LMENTRY we measure performance using the official evaluation script, while in Big-Bench we perform exact string matching. We note that while exact matching is somewhat strict, we believe it is also fair and straightforward.

Model	Model size	Base model	# Params
Flan-T5	Small		80M
	Base		250M
	Large	T5	780M
	XL		3B
	XXL		11B
T0	Small		3B
	T0pp	T5	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.

3.2 Models

We evaluate 16 instruction-tuned LLMs from 11 diverse model families (Chung et al., 2024; Sanh et al., 2021; Taori et al., 2023; Zheng et al., 2024; Durbin, 2023; Ding et al., 2023; NousResearch, 2023; Almazrouei et al., 2023; Team, 2023; Collective, 2023) (see Table 1). We refrain from including 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 thousands of dollars. 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.

4 Evaluating against a Single Prompt Leads to Instability in Results

As discussed in the previous section, LLMs are usually evaluated 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 both absolute and relative model performance.

In Section 4.1 we create a large number of automatically generated instruction paraphrases for tasks from the LMENTRY and BBH benchmarks. Paraphrases are created using an LLM and verified by human annotators. In Section 4.2, we statistically analyze the performance of various LLMs against these instruction templates and quantify the variation in model performance. Finally, in Section 4.3, we show that models exhibit similar brittleness with manually written paraphrases for tasks from the BBL benchmark.

4.1 Paraphrasing Instruction Templates

We use three prompting methods which were found useful in previous works: (1) instruction template rephrasing: asking an LLM to rephrase a seed prompt (Lester et al., 2021; Gonen et al., 2023; Honovich et al., 2023a); (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. (2023b), 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.

In all of the above, we use GPT-3.5-Turbo for generation, and 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. All automatically generated paraphrases were manually verified and filtered by an annotator from our group to ensure their coherence and relevance to the task. A portion of the data involving 15 randomly selected templates from each task, totaling in 375 instructions, was also given to a second annotator; results show reliable agreement (Table 3), indicating our evaluation process is calibrated.

See Table 2 for a fine-grained distribution across the different generation metrics. Overall,

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.

Benchmark	Correct (%)	Agreement (accuracy)	Agreement (Cohen's κ)
LMENTRY	86.0	.953	.774
BBH	86.7	.916	.491

Table 3: Human evaluation of doubly annotated paraphrases. Out of 375 automatically generated instructions, more than 85% were found to be correct by both annotators. Both Cohen's κ and the agreement accuracy indicate varying, yet generally high levels of agreement given pronounced label imbalance.

we found that 90% of the generated paraphrases created for LMENTRY were correct, and roughly 84% of the paraphrases for BBH were correct.

On average, the validation process yields 240 validated instruction paraphrases per task for LMENTRY and 175 paraphrases per task for BBH. Next, we use these paraphrases to quantify performance variability due to instruction template paraphrasing across $\sim 6.5M$ instances.[2]

4.2 Quantifying Performance Variance due to Instruction Paraphrasing

We leverage the collection of validated paraphrases to assess how model performance varies with paraphrasing. Our main finding is that the common approach of evaluating against a single prompt is unstable, leading to unreliable results.

Instance Sampling and Prompt Construction. Our study involves a large number of tasks, models, and instruction paraphrases. However, evaluating LLMs can become prohibitively expensive with the increase of the number of

---

[2]Calculated as the number of models tested per task $\times$ number of paraphrased instructions per task $\times$ 100 samples, across all tasks and benchmarks $\approx 240 \times 16 \times 100 \times 10$ (LMENTRY) $+175 \times 11 \times 100 \times 15$ (BBH).

samples, datasets, models, and instruction templates (Perlitz et al., 2023). To make our evaluation feasible, we chose to 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 ones with the largest number of parameters), and 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), Weber et al. (2023) and Voronov et al. (2024) 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 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 4) 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 the 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 ranking differences in Figure 2. In all three cases, P_1 and P_2 are valid paraphrases, yet they lead to vastly

Tasks	Kendall's W	Friedman p-val
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**
BIG-bench Lite		
known unknown	.316 (weak)	**4.4E-5**
play dialog	.355 (weak)	**4.3E-5**
winowhy	.520 (weak)	**6.0E-4**
strategic qa	.529 (weak)	.191
hindu knowledge	.560 (weak)	.569
conceptual	.731 (medium)	.132
strange stories	.731 (medium)	.431
code desc	.756 (medium)	**.002**
novel concepts	.787 (medium)	.620
logic grid puzzle	.796 (medium)	**.010**
lang. identification	.811 (medium)	**.002**
vitaminc	.888 (strong)	.772
bbq lite	.890 (strong)	**.023**
logical deduction	.913 (strong)	.895

Table 4: 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 represent statistical significance levels that are smaller than 1E-50.

different rankings. For example, T0pp ranks first on the BBH task (center) according to P_1 and only 9th according to P_2. Similarly, Alpaca-13B and Alpaca-7B are in the *top*-performing models on the LMENTRY task P_2, while they rank *last* for P_1.

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
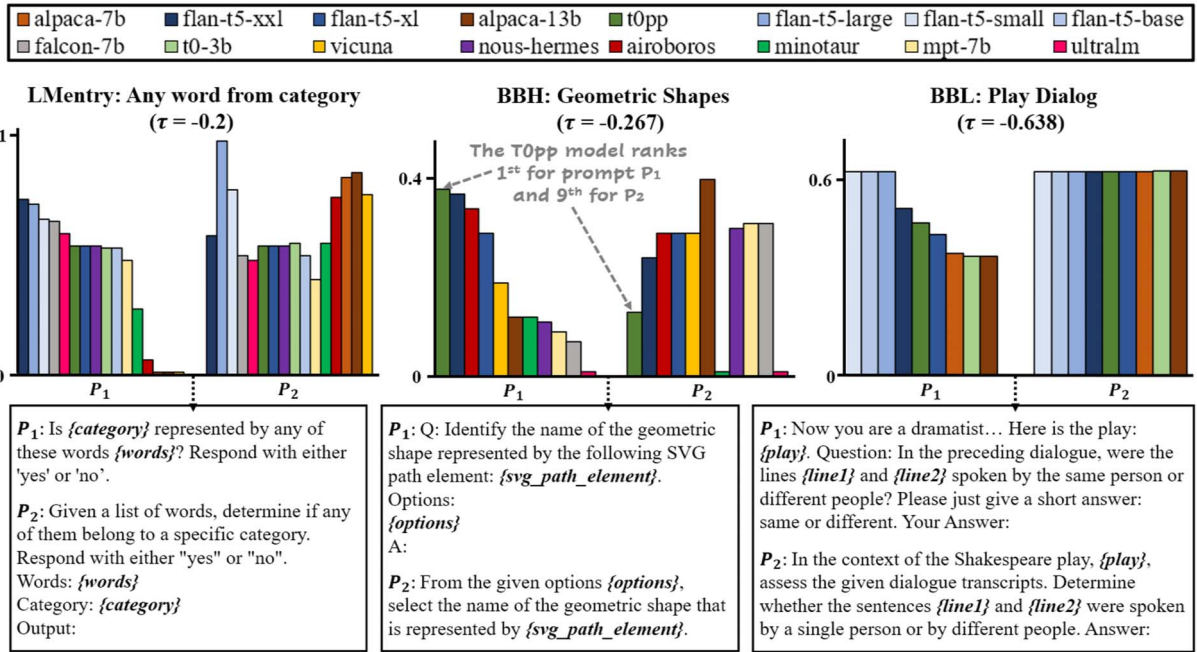
937

Figure 2: Model performance and ranking induced by pairs of paraphrases that exhibit the minimal Kendall τ correlation on three different tasks (one for each benchmark). For each template pair, models are ordered according to their performance against the first instruction template P_1, enabling straightforward comparisons of ranking changes. In other words, if the bars of P_2 appear scattered rather than follow a clear descending order, this indicates a significant reshuffling of rankings.

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). Overall, 15 tasks out of 25 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, deviates from the model's average performance over all paraphrases.

LMentry tasks

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
t03b	-1.97	0.24	0.19	0.67	-0.09	0.35	0.42	-0.05	-0.86	-0.48
t0pp	0.20	-0.03	-0.27	-0.12	0.16	0.28	0.46	-0.23	-0.71	-0.71
fal7b	-0.15	0.04	-0.60	1.08	-0.86	1.17	0.97	0.02	2.40	2.00
mpt7b	-1.98	-2.08	1.30	-0.32	-0.02	-0.04	-0.82	-1.47	-0.84	-0.28
alp7b	1.48	1.74	-0.53	2.14	-0.54	1.79	2.42	1.04	-0.35	-0.44
alp13b	1.80	2.43	0.48	1.91	-0.62	2.15	1.55	1.04	-0.11	1.98
ft5-small	-0.29	0.26	-0.46	-0.79	-0.01	0.16	0.18	-0.22	-0.60	-0.31
ft5-base	-0.53	0.08	-0.82	0.14	-0.57	0.64	0.51	-0.03	-0.89	-0.34
ft5-large	0.58	0.30	-0.64	0.21	0.39	0.16	0.33	-0.04	-1.22	0.73
ft5-xl	1.06	0.71	-0.25	-2.43	-0.39	0.57	0.47	-0.13	-1.73	-0.12
ft5-xxl	0.46	0.56	0.03	0.68	0.05	0.67	0.79	-0.08	-3.77	0.09
air13b	0.41	1.34	0.74	0.28	0.03	1.10	1.20	-0.21	0.29	-0.51
nou13b	0.59	0.23	0.26	-0.09	-0.68	1.24	1.23	-0.73	-0.52	0.92
ult13b	0.41	-0.09	-0.35	0.90	0.11	1.50	1.08	-0.48	-0.18	0.38
vic13b	1.60	2.19	0.52	0.82	-0.52	2.53	1.97	-0.14	0.02	0.88
min15b	-0.74	-1.55	-0.14	0.22	-0.42	1.78	1.59	-0.57	-0.51	0.45

Figure 3: Model and task performance divergence. For each LMENTRY task, we show the number of standard deviations by which performance of each model on the original instructions deviates from averaged performance. Dark cells indicate substantial divergence values (>1 std).

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

Change	t03b	t0pp	fal7b	mpt7b	al7b	al13b	ft5s	ft5b	ft5l	ft5xl	ft5xxl	ai13b	no13b	ul13b	vi13b	mi15b
+ '.'	−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
. -> :	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
… -> ''…''	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
… -> '…'	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
+ 'Q:'	−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'	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'	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'	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
create -> write	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
create -> generate	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	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	−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	−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	0.01	0.02	0.00	0.01	0.00	−0.02	0.23	0.13	0.10	0.03	−0.03	−0.02	0.01	0.04	−0.02	0.02
a -> the	0.00	0.01	0.02	0.15	0.01	0.02	0.03	0.00	0.03	−0.03	0.04	−0.02	0.01	0.04	0.02	0.00
lesser -> smaller	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	−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	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	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	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	−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	−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 4: Average performance differences between various models for the most common minimal edits between two instruction templates (e.g., substituting 'excludes' with 'lacks') in the LMENTRY benchmark.

the Alpaca-13B with the original instruction templates outperformed its average performance by more than one standard deviation in 7 out of 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 Figure 4 and Table 5.

4.3 LLMs are Also Sensitive to Manual Paraphrases

Inconsistencies observed in our analyses could stem from paraphrases that leaked to the training of the models. To address this, we extended our analysis with instruction paraphrases which were recently written by Sun et al. (2023) for the BBL tasks (7–12 instruction templates per task). Importantly, these human-crafted paraphrases were written *after* model training.

We use these annotations to examine model performance. Our analysis revealed similar inconsistencies as observed with automated paraphrases, demonstrating model sensitivity to paraphrasing even when the potential for instruction leakage is minimized. See Table 4 for the Kendall's W values for all BBL tasks, and Figure 2 for a pair of instruction templates exhibiting the minimal Kendall's τ correlation across all BBL tasks.

5 Different Use Cases Merit Different Metrics

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 $\varepsilon(M, T, i) \in [0, 1]$

939

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	+.61
	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
	flan-t5-xl	Write a word that omits the letter ''letter''	.77	Write a word that omits the letter ''letter''.	.54	−.23
+ '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
omits –> lacks	ultralm-13b	Write a word that ***omits*** the letter ''{letter}''.	.62	Write a word that ***lacks*** the letter ''{letter}''.	.19	−.42
	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
	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
	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

Table 5: Representative examples of instruction template pairs from LMENTRY with very minor differences but notable variations in performance (open-source models).

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 down-

stream task and domain (e.g., 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, we advise developers to use a new sample set for the task. This ensures the chosen prompt is validated by its ability to maximize performance on these held-out samples irrespective of prior exposure during training.

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 (Le Scao et al., 2022), as it mitigates outliers in performance.

5.3 Combined Performance Score

In the same way that 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 robustness 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 robust (high Sat). CPS facilitates identifying models that strike a balance between top-tier performance and robust reliability across varying instruction templates.

6 Multi-Prompt Evaluation

In Figure 6 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
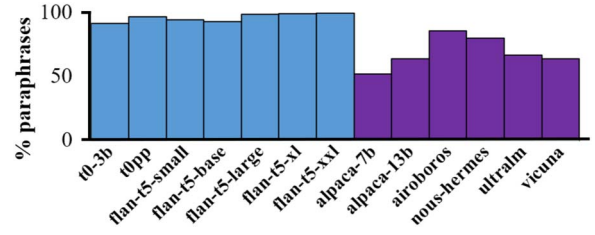

Figure 5: Percentage of instruction paraphrases with accuracy higher than 5% in T5 models (blue) vs. LLaMA models (purple) on LMENTRY tasks.

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

More broadly, model ranking depended on the metric used. For instance, see Figure 6 (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 %paraphrases that achieved at least 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 filtering of paraphrases. The results (Table 6) 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
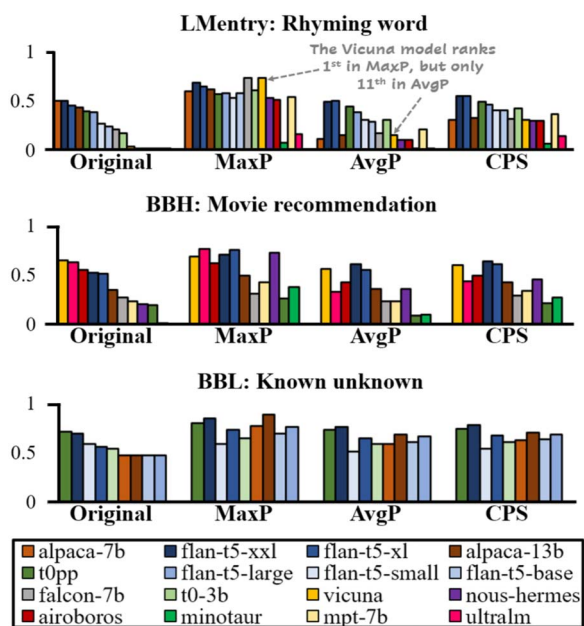
Figure 6: The performance of various models according to the metrics proposed in Section 5, evaluated on sample tasks from each of the three benchmarks. The name of the metric appears below each group of columns; height of a column represents value in *that specific metric*. The order of the columns (i.e., models) between groups is fixed, set according to decreasing performance on the original instruction templates to enable straightforward comparisons of ranking changes.

Benchmark	Max perf.	Average perf.	Combined perf.
LMENTRY	.963	.978	.948
BBH	.991	.983	.966

Table 6: Averaged Kendall's τ values comparing rankings before and after filtering incorrect paraphrases for each metric across all tasks (excluding ''ends with word'' for LMENTRY).

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.

7.1 Results

Below we summarize the results of our evaluation of OpenAI models. The full details appear in our repository.

OpenAI Models are Also Sensitive to Minor Prompt Variations. Minor changes in the phrasing of the instruction could lead to drastic performance changes for the OpenAI models, similar to our findings in Section 4.2 with smaller-scale LLMs. See representative examples in Table 7, showing nearly identical instruction template pairs resulting in notable variations in performance.

Average Performance is Lower Than That Observed in the Original Benchmark Instructions. In 72.5% of the cases, the performance of the original instructions was higher than the estimated average across all paraphrases. In the davinci model, the original prompts added on average 21 more accuracy points.

Original Prompt Performances Fall Below All Paraphrases' Estimated Maximum Performance. Figure 7 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

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 7: Minimal distance pairs from LMENTRY with large performance differences in OpenAI models.
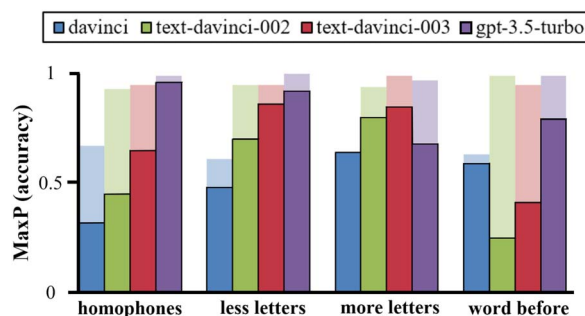
Figure 7: Comparison of the *maximum performance* of four OpenAI models using original prompts (in solid colors) vs. all prompt paraphrases (semi-transparent). Each group of columns corresponds to a different task in the LMENTRY benchmark.

8 out of 10 tasks. Across all four models, 26 out of 40 differences were statistically significant according to the McNemar test.

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 developed methods for choosing subsets of the test data which are expected to be

943

a good representative of the whole. An interesting avenue for future work can extend Perlitz et al.'s (2023) 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 7). Voronov et al. (2024) showed that LLMs are sensitive to the format of in-context examples. For example, they varied 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.

The works discussed above represent a distinct thread within the larger field of model robustness, which is typically defined as a measure of models' ability to adapt to distribution shifts between training and inference (Wang et al., 2022), or to cope with adversarial examples (Wang et al., 2021, 2023). In contrast, these works do not change the underlying instance to be classified (e.g., the homophone pairs in our running example), but rather the task *instruction*. This challenge arises with the introduction of LLMs which take such instructions as part of the input, rather than through dedicated calibration in training or finetuning.

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.

Acknowledgments

References

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

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, Technology Innovation Institute.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben

Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

OpenAccess AI Collective. 2023. Minotaur. https://huggingface.co/openaccess-ai-collective/minotaur-15b. Last Accessed: 2024-04-30.

Gregory W. Corder and Dale I. Foreman. 2011. *Nonparametric Statistics for Non-Statisticians*. John Wiley & Sons, Inc.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051.

https://doi.org/10.18653/v1/2023.emnlp-main.183

Jon Durbin. 2023. Airoboros. https://github.com/jondurbin/airoboros. Last Accessed: 2024-04-30.

Avia Efrat, Or Honovich, and Omer Levy. 2023. Lmentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501. https://doi.org/10.18653/v1/2023.findings-acl.666

Gemini Team Google, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng,z HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv: 2312.11805*.

945

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148. https://doi.org/10.18653/v1/2023.findings-emnlp.679

Jiasheng Gu, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin. 2023. Robustness of learning from task instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13935–13948. https://doi.org/10.1016/j.learninstruc.2022.101692

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023a. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428. https://doi.org/10.18653/v1/2023.acl-long.806

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023b. Instruction induction: From few examples to natural language task descriptions. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 1935–1952. Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2023.acl-long.108

Leonard J. Kazmier, Michael K. Staton, and Daniel L. Fulks. 2003. Business statistics: Based on schaums outline of theory and problems of business statistics, by Leonard J. Kazmier, McGraw-Hill.

Maurice G. Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251. https://doi.org/10.1093/biomet/33.3.239, PubMed: 21006841

Maurice G. Kendall and B. Babington Smith. 1939. The problem of m rankings. *The Annals of Mathematical Statistics*, 10(3):275–287. https://doi.org/10.1214/aoms/1177732186

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv e-prints*, pages arXiv–2211.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.243

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro

Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

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. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 3470–3487. Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2022 .acl-long.244

NousResearch. 2023. Nous-hermes. https:// huggingface.co/NousResearch/Nous -Hermes-13b. Last Accessed: 2024-04-30.

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 preprint arXiv:2308.11696*.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler,

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, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

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. In *The Twelfth International Conference on Learning Representations*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt,

Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Cristina Garbacea, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research.*

Jiuding Sun, Chantal Shaib, and Byron C. Wallace. 2023. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations.*

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051. https://doi.org/10.18653/v1/2023.findings-acl.824

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. Last Accessed: 2024-04-30.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. www.mosaicml.com/blog/mpt-7b. Last Accessed: 2024-04-30.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 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. *arXiv preprint arXiv:2401.06766.*

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).*

Jindong Wang, Hu Xixu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxin Jiao, Yue Zhang, and Xing Xie. 2023. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models.*

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in nlp models: A survey. In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 4569–4586. Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2022.naacl-main.339

Lucas Weber, Elia Bruni, and Dieuwke Hupkes. 2023. Mind the instructions: A holistic evaluation of consistency and interactions in prompt-based learning. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 294–313. https://doi.org/10.18653/v1/2023.conll-1.20

Jason Wei, Maarten Bosma, Vincent 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. In *International Conference on Learning Representations.*

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

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