P-MMEVAL: A Parallel Multilingual Multitask Benchmark for Consistent Evaluation of LLMs

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs) showcase varied multilingual capabilities across tasks like translation, code generation, and reasoning. Previous assessments often limited their scope to fundamental natural language processing (NLP) or isolated capability-specific tasks. To alleviate this drawback, we aim to present a comprehensive multilingual multitask benchmark. First, we present a pipeline for selecting available and reasonable benchmarks from massive ones, addressing the oversight in previous work regarding the utility of these benchmarks, i.e., their ability to differentiate between models being evaluated. Leveraging this pipeline, we introduce P-MMEVAL, a large-scale benchmark covering effective fundamental and capability-specialized datasets. Furthermore, P-MMEVAL delivers consistent language coverage across various datasets and provides parallel samples. Finally, we conduct extensive experiments on representative multilingual model series to compare performances across models, analyze dataset effectiveness, examine prompt impacts on model performances, and explore the relationship between multilingual performances and factors such as tasks, model sizes, and languages. These insights offer valuable guidance for future research [1].

1 Introduction

In recent years, large language models (LLMs, Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023; Bai et al., 2022, 2023) have raised significant interest in the artificial intelligence (AI) community. As most LLMs are English-centric, when we focus on the performances of a specific LLM, it generally refers to the evaluation results on English benchmarks. For example, early research focuses on reporting evaluation results on fundamental natural language processing (NLP) benchmarks. i.e, how accurately the LLM understands and generates text, including TRIVIAQA (Joshi et al., 2017a), WINOGRANDE (Sakaguchi et al., 2020), and HELLASWAG (Zellers et al., 2019). Nowadays, researchers are more interested in capability-specialized benchmarks, i.e., how well LLM performs on a group of specific task-solving problems, including GSM8K (Cobbe et al., 2021) for mathematical reasoning, MMLU (Hendrycks et al., 2021a) for knowledge acquisition, and HUMANEVAL (Chen et al., 2021) for code generation. However, there is currently little work on systematically evaluating the multilingual capabilities of LLMs. When developing and iterating LLMs, giving accurate and parallel evaluation results is crucial for identifying their multilingual capabilities and quantifying their performances.

Building a benchmark with both inclusive task coverage and strong linguistic parallelism is difficult. Measuring the multilingual abilities of a specific LLM, or comparing the quality of generated multilingual responses from one LLM to another, remains a big challenge in developing multilingual LLMs. Early work focuses on an isolated evaluation pipeline for a specific task, or to be more concrete, a specific perspective of LLM abilities: MHELLASWAG (Dac Lai et al., 2023) aims at collecting the multilingual understanding abilities, XLSUM (Hasan et al., 2021) mainly focus on evaluating the quality of generated multilingual text, HUMANEVAL-XL (Peng et al., 2024) is used for quantify how well-executed the generated code segments are, and MGSM (Shi et al., 2023) is made for testifying the performance on arithmetic reasoning. In modern research, for delivering simpler aggregation and comprehensive evaluation when judging model abilities, researchers collect several popular isolated benchmark tasks and propose a united, large-scale multilingual benchmark system like XTREME (Hu et al., 2020), XTREME-

[1] We will publish all the code and resources after the paper is received.

R (Ruder et al., 2021), XGLUE (Liang et al., 2020), MEGA (Ahuja et al., 2023), and BUFFET (Asai et al., 2024) for multi-task assessments. However, these large-scale benchmarks 1) are tailored predominantly to fundamental NLP tasks and 2) inconsistently cover multiple languages across their selected datasets.

In this paper, our goal is to present a pipeline to develop a comprehensive multilingual multitask benchmark. To this end, we first select representative and challenging datasets from fundamental NLP tasks to reduce redundant testing and enhance the efficiency of evaluation. The second phase of our endeavor involves a meticulous curation of the most intensely studied capability-specialized tasks in contemporary research including code generation, knowledge comprehension, mathematical reasoning, logical reasoning, and instruction following. Finally, we construct a collection of datasets P-MMEVAL, consisting of three fundamental NLP datasets and five advanced capability-specialized datasets. To maintain language coverage among all selected datasets, we unify 10 languages considering the cost and computational limitations via expert translation review to construct the missing multilingual portions.

To summarize, our contributions are as follows:

- We present a pipeline for selecting available and reasonable benchmarks to assess the multilingual abilities of LLMs. Innovatively, we employ a statistical analysis method to identify effective datasets from a collection of datasets. Our method can enhance the objectivity and scientific rigor of the selection process.

- We develop a multilingual multi-task benchmark P-MMEVAL that includes both fundamental and capability-specialized tasks, which ensures consistent language coverage across various datasets and provides parallel samples across different languages. This benchmark facilitates a thorough assessment of multilingual capabilities and enables unprecedented fairness and consistency in evaluating cross-lingual transfer capabilities.

- Our experiments offer a comprehensive analysis of the multilingual capabilities of various LLMs, showcasing performance across different prompts, models, languages, and tasks. Importantly, we analyze the utility of each dataset within P-MMEVAL in distinguishing model performance, thus identifying specific benchmarks that differentiate model performance across model series and sizes.

2 Related Work

Isolated Fundamental NLP Benchmarks Although diverse multilingual evaluation benchmarks have been established, they focused on basic language understanding and generation capabilities of models. Notable work includes XNLI (Conneau et al., 2018) dataset for natural language inference, XCOPA (Ponti et al., 2020), MHELLASWAG (Dac Lai et al., 2023), and XWINOGRAD (Tikhonov and Ryabinin, 2021) for commonsense reasoning, PAWS-X (Yang et al., 2019) for paraphrase identification, XL-WIC (Raganato et al., 2020) for word sense disambiguation, as well as the span extraction QA datasets including XQUAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), and TYDIQA-GOLDP (Joshi et al., 2017b). Additional examples include XLSUM (Hasan et al., 2021) for text summarization and FLORES-200 (Costa-jussà et al., 2022) for machine translation. Each of those benchmarks is typically designed for a specific task, solely focusing on one aspect of the model's capabilities.

Unified Fundamental NLP Benchmarks There are also large-scale benchmarks that unify diverse existing datasets, aiming at offering a comprehensive evaluation of the model's abilities from various perspectives. For instance, XTREME (Hu et al., 2020) comprises four tasks related to natural language understanding (NLU). Its refined version, XTREME-R (Ruder et al., 2021), optimizes the specific datasets tailored for each task category within XTREME. The XGLUE (Liang et al., 2020), MEGA (Ahuja et al., 2023), and BUFFET (Asai et al., 2024) benchmarks integrate various datasets for both understanding and generation tasks. The BUFFET benchmark also provides a fixed set of few-shot demonstrations for evaluation.

Capability-specialized Multilingual Benchmarks The advanced task-solving capabilities of LLMs have garnered significant attention from the research community. The six capabilities that receive the most emphasis are mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021b), logical reasoning (Liu et al., 2020), instruction following (Li et al., 2023), knowledge

Source	Task	Benchmarks	# Examples	Test sets	Metric
Existing	Generation	FLORES-200 (Costa-jussà et al., 2022)	1012 × 10	Annotation	BLEU
Extension	Understanding	XNLI (Conneau et al., 2018)	120 × 10 (3)	Translation	Acc
		MHELLASWAG (Dac Lai et al., 2023)	120 × 10 (3)	Translation	Acc
	Code generation	HUMANEVAL-XL (Peng et al., 2024)	80 × 10 (3) × 12	Translation	Pass@1
	Mathematical reasoning	MGSM (Shi et al., 2023)	250 × 10 (3)	Translation	Acc
	Logic reasoning	MLOGIQA (Liu et al., 2020)	80 × 10 (8)	Translation	Acc
	Knowledge	MMMLU (Hendrycks et al., 2021a)	400 × 10 (2)	Translation	Acc
	Instruction following	MIFEVAL (Zhou et al., 2023)	96 × 10 (9)	Translation	Acc

Table 1: An overview of the P-MMEVAL benchmark. In total, P-MMEVAL takes seven multilingual tasks into consideration, which is built on eight benchmarks. "# Examples" denotes "the number of examples per language" × "the number of involved languages" × "the number of programming languages" (special for HUMANEVAL-XL), and the numbers of extended languages are in parentheses. "Test sets" section describes the nature of the test sets (whether they are translations of English data or independently annotated).

comprehension (Hendrycks et al., 2021a), code generation (Chen et al., 2021), and conversational abilities (Bai et al., 2024). Typical multilingual benchmarks include MGSM (Shi et al., 2023) for mathematical reasoning, the OpenAI multilingual version of MMLU (MMMLU)[2] for knowledge comprehension, and HUMANEVAL-XL (Chen et al., 2021) for code generation.

All the benchmarks mentioned above focus either exclusively on fundamental NLP capabilities or on advanced application abilities. Additionally, there is inconsistent multilingual coverage across various datasets within a single multi-task benchmark. The proposed benchmark P-MMEVAL integrates three fundamental NLP datasets and five capability-specialized datasets, providing consistent language coverage across all selected datasets.

3 Datasets Selection Pipeline

Through the accumulation of a long time, the evaluation tasks for language models encompass a wide variety, with each category amassing substantial multilingual datasets. These datasets are primarily categorized into two main types: generation and understanding. Each task is further divided into various subcategories, most of which consist of multiple datasets. Therefore, selecting effective ones is crucial, as it can reduce redundant testing and improve evaluation efficiency. To achieve this, we utilize paired-sample T-test (Field, 2005) to optimize the selection process by filtering out datasets that can effectively distinguish the performances of LLMs among different model series and sizes. We suggest that if these benchmarks do not maintain significant differences even when the size gap is large enough, their evaluation results can be considered ineffective. Therefore, those benchmarks can not present reliable and meaningful performance identification and comparison.

Our selection pipeline can be described as follows: Given the evaluation results of model A and model B on a multilingual dataset D, denoted as A_i and B_i respectively, where i represents the language index. Following this, we first collect two score arrays $[A_1, A_2, ..., A_m]$ and $[B_1, B_2, ..., B_m]$ which represents the evaluation results of model A and model B on m different languages, respectively. Then, we use these two arrays to derive the significance value p after running a paired-T significance test. If p is less than a pre-defined significance level (e.g., 0.01), it can be concluded that there is a significant difference in the overall scores between model A and model B. By determining whether multiple pairs of models have significantly different scores on this dataset, the effectiveness of the dataset in distinguishing the performance among various models can be identified.

4 P-MMEval

We aim to build a comprehensive evaluation system that unifies diverse NLP and capability-specialized tasks, ensures consistent language coverage per task, and offers parallel samples across languages to facilitate consistent comparisons. The overview of our proposed P-MMEVAL benchmark is shown in Table 1.

[2] https://huggingface.co/datasets/openai/MMMLU

4.1 Design Principles

Diversity in tasks First, the two key fundamental NLP tasks of generating and understanding are covered. More critically, through in-depth analysis, we identify and establish five kinds of core capabilities of current LLMs, including code generation, knowledge comprehension, mathematical reasoning, logical reasoning, and instruction following.

Diversity in languages To ensure that our benchmark can also help testify the cross-lingual transferability of LLMs, we unify 10 different languages spanning 8 language families, including English (*en*), Chinese (*zh*), Arabic (*ar*), Spanish (*es*), Japanese (*ja*), Korean (*ko*), Thai (*th*), French (*fr*), Portuguese (*pt*), and Vietnamese (*vi*).

4.2 Fundamental NLP Dataset Curation

In light of the diversity of fundamental NLP datasets, we meticulously select 11 datasets widely employed in research (Ahuja et al., 2023; Asai et al., 2024; Liang et al., 2020), spanning across the two major categories of understanding and generation. This curation aims to thoroughly appraise the models' foundational capabilities. Below, we briefly summarize these two categories of tasks.

4.2.1 Tasks

Natural Language Understanding (NLU) Here, we have five different sub-tasks: i) The natural language inference (NLI) dataset, XNLI (Conneau et al., 2018), which involves classifying whether a hypothesis is entailed, contradicted, or unrelated to the premise. ii) Three commonsense reasoning datasets encompass XCOPA (Ponti et al., 2020) focusing on causal reasoning, MHELLASWAG examining social scenarios and linguistic fluency, and XWINOGRAD (Tikhonov and Ryabinin, 2021) addressing anaphora resolution issues. iii) The paraphrase identification dataset PAWS-X (Yang et al., 2019) requires the model to determine whether two given sentences convey the same meaning. iv) The word sense disambiguation dataset XL-WIC (Raganato et al., 2020) focuses on understanding the meanings of words in various contexts. v) Three span-prediction datasets, i.e., XQUAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), and TYDIQA-GOLDP (Joshi et al., 2017b), where the answer to a question is provided within a piece of context.

Natural Language Generation (NLG) This task comprises the XLSUM (Hasan et al., 2021) and FLORES-200 (Costa-jussà et al., 2022) datasets. XLSUM is a multilingual summarization dataset derived from news articles. FLORES-200 is a dataset for multilingual machine translation, covering 200 languages.

4.2.2 Settings

We utilize three pairs of models to help fundamental benchmark curation, including QWEN2.5-7B vs. QWEN2.5-72B (Yang et al., 2024), LLAMA3.1-8B vs. LLAMA3.1-70B (Dubey et al., 2024), and MISTRAL-NEMO-INSTRUCT-2407 (MISTRAL-NEMO) vs. MISTRAL-LARGE-INSTRUCT-2407 (MISTRAL-LARGE).[3] For understanding tasks, we utilize a fundamental prompt design with English instructions (See "EN" format in Section 5.2). For generation tasks, we employ the native prompt with instructions in the target language (See "Native" format in Section 5.2), as the "EN" prompt can cause the model to generate responses in English for non-English data. Then, we count the number of occurrences of each language in all benchmarks. For each benchmark, aside from English, we select four extra languages that are both supported in that benchmark and deserve the highest occurrences in all benchmarks. To expedite result verification, we gather a maximum of 250 instances per language across all tasks, ensuring an efficient yet comprehensive evaluation process.

4.2.3 Results

Table 2 presents the paired-sample T-test results, identifying significant differences in pairwise model performances on each dataset. The p-value threshold is set at 0.01. The dataset will be retained if all three selected model pairs show significant performance differences. Following this criterion, XNLI, MHELLASWAG, and FLORES-200 are retained for further processing and extension.

4.3 Capability-specialized Dataset Curation

Besides the fundamental NLP tasks mentioned above, we also select one dataset for each of the five capability-specialized tasks.[4] To maintain consistency across all languages, we extend the support of some benchmark datasets on the missing languages by collecting human-annotated translation results. We first deliver the translated examples

[3] https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407 and https://huggingface.co/mistralai/Mistral-Large-Instruct-2407.

[4] For each specialized capability, we generally do not have enough choices (mostly only one benchmark is available).

Dataset	Available	Model series		
		QWEN	LLAMA	MISTRAL
Understanding				
XNLI	✓	0.0055	0.0009	0.0005
MHELLASWAG	✓	0.0028	0.0078	0.0039
PAWS-X	✗	0.5794	0.0170	0.0008
XL-WIC	✗	0.1734	0.0078	0.0058
XCOPA	✗	0.0070	0.0110	0.0014
XWINOGRAD	✗	0.0224	0.0002	0.0014
XQUAD	✗	0.0283	0.0066	0.0117
TYDIQA-GOLDP	✗	0.2494	0.0375	0.0001
MLQA	✗	0.0011	0.0710	0.0064
Generation				
FLORES-200	✓	0.0010	0.0031	0.0007
XLSUM	✗	0.4835	0.7518	0.1500

Table 2: Results on significance test among three pairs of models: QWEN2.5-7B/72B (QWEN), LLAMA3.1-8B/70B (LLAMA), and MISTRAL-NEMO/LARGE (MISTRAL). For the understanding task and the generation task, we finally select XNLI and MHELLASWAG, and FLORES-200, respectively, as their significance level values are all lower than 0.01.

generated by powerful LLM, and require a professional translation team to conduct a thorough review of the machine translation results, correct translation errors if necessary, localize vocabulary expressions, and eliminate cases that cannot be directly mapped across languages, thus ensuring translation quality and cultural adaptability (See Table 6). In detail, the involved specialized capabilities in P-MMEVAL are:

- **Code generation** We utilize HUMANEVAL-XL (Peng et al., 2024) dataset, which establishes connections between 23 natural languages (NLs) and 12 programming languages (PLs). We collect 80 examples in *ja*, *ko*, and *th* in extension.
- **Mathematical reasoning** We use the MGSM (Shi et al., 2023) dataset, a multilingual version translated from the monolingual GSM8K dataset consisting of math word problems. We extend its multilingual support with *ar*, *ko*, *pt*, and *vi* examples.
- **Logical reasoning** We keep the original *en* and *zh* examples from origin LOGIQA (Liu et al., 2020) dataset. Besides, we extend its multilingual version by translating *en* examples into *ar*, *es*, *ja*, *ko*, *th*, *fr*, *pt*, and *vi*.
- **Knowledge aqcuisition** We sample a subset of MMMLU comprising 200 "hard" samples and 200 "easy" samples. The performance of six diverse models (QWEN2.5-7B, QWEN2.5-72B, LLAMA3.1-8B, LLAMA3.1-70B, MISTRAL-NEMO, and MISTRAL-LARGE) is utilized as a proxy for selecting "hard" and "easy" samples. Concretely, we compile an "easy" subset comprising 6,335 instances where all models excel, and a "hard" subset consisting of 663 instances that challenge every model. Subsequently, guided by annotations from MMLU-REDUX (Gema et al., 2024), we refine these subsets by discarding 798 erroneous instances from the "easy" pool and 160 from the "hard" pool. Finally, we systematically sample 200 instances from each of the pruned pools, thus creating our finalized "easy" and "hard" evaluation sets. We translate those examples into *th* and *fr*.
- **Instruction following** We employ the English IFEVAL (Liu et al., 2020) dataset, which consists examples following pre-defined 25 types of "verifiable instruction". We also extend its multilingual version MIFEval with the support in *zh*, *ar*, *es*, *ja*, *ko*, *th*, *fr*, *pt*, and *vi*, where 96 examples for each language.

4.4 Instruction selection

We utilize English instructions from OPENCOMPASS (Contributors, 2023) and LM-EVALUATION-HARNESS (Dac Lai et al., 2023). Among multiple instructions, we select a suitable one and make uniform modifications to ensure consistency across similar tasks. For zero-shot prompts, to increase the success rate of answer extraction, we add a constraint at the end of the instruction to some tasks, requiring the model to output the generated answers in a fixed format. In addition, we translate English instructions into multiple languages to construct native instructions.

5 Experiments

This section focuses on the following aspects: assessing the multilingual capabilities of different models; assessing the utility of each dataset within P-MMEVAL in distinguishing model performance; examining the influence of various prompts on multilingual performance; and analyzing the correlation between models' performance in English and non-English languages. All evaluation results are conducted in Table 3.

Model	Understanding		Code generation	Mathematical reasoning	Logic reasoning	Knowledge	Instruction following	Generation	AVG_S	AVG_U
	XNLI	MHELLASWAG	HUMANEVAL-XL	MGSM	MLOGIQA	MMMLU	MIFEVAL	FLORES-200		
Open-source models (<7B)										
LLAMA3.2-1B	31.67	24.49	37.71	12.08	27.12	27.80	35.42	29.30	28.03	28.08
LLAMA3.2-3B	30.67	23.74	37.42	11.64	25.62	26.85	34.90	**36.85**	27.29	27.21
QWEN2.5-0.5B	22.25	19.68	33.92	13.12	14.62	30.25	30.21	15.95	24.42	20.97
QWEN2.5-1.5B	46.58	36.35	48.59	35.20	35.12	42.02	44.37	21.37	41.06	41.47
QWEN2.5-3B	60.08	48.09	60.75	69.40	39.38	46.27	66.46	25.75	**56.45**	**54.09**
GEMMA2-2B	53.50	45.31	51.54	44.52	34.88	40.85	56.67	24.00	45.69	49.41
Open-source models (7-14B)										
LLAMA3.1-8B	52.84	49.11	69.96	67.24	39.88	43.80	59.27	16.59	56.03	50.98
QWEN2.5-7B	67.17	62.92	71.88	81.08	45.88	49.83	77.71	32.76	65.28	65.05
GEMMA2-9B	57.92	65.62	69.96	81.28	41.50	49.23	79.17	**36.48**	64.23	61.77
MISTRAL-NEMO	54.25	55.73	57.38	76.52	41.75	44.88	60.00	33.65	56.11	54.99
QWEN2.5-14B	67.50	70.10	72.83	88.68	53.50	51.52	79.48	31.31	**69.20**	**68.80**
Open-source models (14-50B)										
QWEN2.5-32B	68.33	76.38	75.88	90.88	57.38	52.27	83.33	32.13	**71.95**	**72.36**
GEMMA2-27B	68.00	64.12	76.67	85.28	50.50	49.42	81.35	**42.23**	68.64	66.06
Open-source models (>50B)										
LLAMA3.1-70B	63.17	67.25	74.75	88.28	52.38	55.52	79.17	16.63	70.02	65.21
QWEN2.5-72B	71.42	75.95	76.00	91.00	58.38	52.67	87.60	41.55	**73.13**	**73.69**
MISTRAL-LARGE	69.58	69.04	77.17	90.48	53.50	51.85	83.23	**43.40**	71.25	69.31
Close-source models										
GPT-4O	69.17	81.04	77.05	91.60	56.75	55.77	85.21	**46.32**	73.28	**75.11**
CLAUDE-3.5-SONNET	71.50	77.72	82.92	92.84	62.25	56.17	80.73	16.20	**74.98**	74.61

Table 3: Evaluation results of different models on P-MMEVAL. We gather those models by referring to their sizes. AVG_U and AVG_S represent the average score of the understanding and capability-specialized tasks, respectively. HUMANEVAL-XL score presents the average score of three programming languages.

5.1 Multilingual Models

We evaluate the performance of several representative instruction-tuned models – (i) closed-source models GPT-4O[5] (OpenAI, 2023) and CLAUDE-3.5-SONNET[6], (ii) open-source models including LLAMA3.1, LLAMA3.2 (Dubey et al., 2024), QWEN2.5 (Yang et al., 2024), MISTRAL-NEMO, MISTRAL-LARGE, and GEMMA2 series (Rivière et al., 2024).

5.2 Evaluation Settings

According to Zhao et al. (2021), the choice of prompts significantly impacts the evaluation results of LLMs and the model performance is sensitive to minor variations in prompting. In this study, we compare the evaluation results using the following prompts:

- EN: Instructions in English + input in the target language.
- Native: Instructions in the target language + input in the target language.
- EN-Few-Shot: Instructions in English + demonstrations in the target language + input in the target language.

For MGSM, we employ Chain of Thought (CoT) (Wei et al., 2022) reasoning, which guides the model to think step-by-step before providing a final answer. For XNLI, MHELLASWAG, MLOGIQA, HUMANEVAL-XL, MIFEVAL, and FLORES-200, direct answering is utilized, which requests the model to produce answers directly. The inference methods for these datasets align with the most commonly used settings. Notably, for MMMLU, we choose the prompt template following OpenAI `simple-evals` repository.[7] Specifically, CoT reasoning exhibits a significantly higher answer extraction failure rate compared to direct answering on small-sized LLMs (i.e., the number of parameters is less than 7B), leading to poor performance. Thus, we employ a direct answering prompt for small-sized LLMs. The detailed evaluation prompts are illustrated in Appendix G.

For the few-shot demonstrations, we primarily sample demonstrations from the validation set of the original dataset. For the missing multilingual portions, we utilize GPT-4O to translate these demonstrations from English into the missing languages. Please note that the demonstrations serve only as an answer format.

5.3 Main Results

Table 3 presents an overview of the evaluation results. Unless otherwise noted, the standard EN prompt is applied to all datasets except FLORES-

[5] `gpt-4o-2024-05-13`

[6] `claude-3-5-sonnet-20240620`

[7] https://github.com/openai/simple-evals

200, HUMANEVAL-XL, and MIFEVAL, where the Native prompt is required. More information about the prompting strategies including EN, Native, and En-Few-Shot is shown in Appendix A. The evaluation result on HUMANEVAL-XL is the average score across three programming languages including Python, JavaScript, and Java. See Appendix C for programming language evaluation details.

First, the multilingual capabilities of models become stronger as the model sizes increase (Kaplan et al., 2020). One exception is that when the size of LLAMA3.2 increases from 1B to 3B, there is a slight decline in performance. The main reason for this is that LLAMA3.2-1B and LLAMA3.2-3B exhibit poor instruction-following capabilities, leading to a higher failure rate in answer extraction and, consequently, fluctuations in the final score. As the model size increases, the improvements in various multilingual tasks show significant differences. Evaluation results on the understanding and capability-specialized tasks show significant improvement in understanding context, processing semantic information, reasoning, and special abilities, with increasing model sizes. For example, for the QWEN2.5 series, the scores on the MGSM dataset for the 0.5B and 72B models are 13.12 and 91.00, respectively. In contrast, the models' performance on generation tasks is relatively weaker and shows slight improvement. Evaluations on the FLORES-200 datasets indicate that, despite the increase in model size, the generation capability does not improve proportionally. This may reflect the complexity of generating text that maintains logical coherence and contextual relevance, where increasing model sizes does not significantly enhance output quality.

In addition, QWEN2.5 demonstrates a strong multilingual performance on understanding and capability-specialized tasks, while GEMMA2 excels in generation tasks. CLAUDE-3.5-SONNET performs poorly on FLORES-200 because it tends to generate additional relevant statements in its responses, potentially downgrading the BLEU score. GPT-4O generally outperforms open-source models. The performance gap between the best-performing open-source model and GPT-4O is within 3%.

6 Analyses

6.1 Analysis on Dataset Utility

The primary objective of this section is to assess the utility of each dataset within P-MMEVAL in distinguishing model performances. We divide open-sourced models into categories by two aspects: model series and model sizes. Specifically, we collect 5 categories of models from 5 model series:

- QWEN2.5: 0.5B, 1.5B, 3B, 7B, 14B, 32B, 72B
- LLAMA3.1: 8B, 70B
- LLAMA3.2: 1B, 3B
- GEMMA2: 2B, 9B, 27B
- MISTRAL: NEMO, Large

And, we divide them into three categories based on their sizes:

- Less than 7B (<7B): QWEN2.5-0.5B, QWEN2.5-1.5B, QWEN2.5-3B, LLAMA3.2-1B, LLAMA3.2-3B, GEMMA2-2B
- Between 7B and 14B (7B-14B): QWEN2.5-7B, LLAMA3.1-8B, GEMMA2-9B, MISTRAL-NEMO, QWEN2.5-14B
- Larger than 70B (>70B): LLAMA3.1-70B, QWEN2.5-72B, MISTRAL-LARGE

Table 4 shows the utility of each dataset in distinguishing the performances of paired models within the same category. The detailed method for calculating the utility of each dataset is presented in Appendix E. A value closer to 1 indicates higher utility for the dataset, with a value of 1 signifying that all models within the same category demonstrate distinguishable performances. Conversely, a numerator of 1 indicates that no models are distinguishable on that dataset. We set the utility threshold at 0.5, where each value is considered effective or ineffective in distinguishing the performances of models with the specified dataset. Based on the results in Table 4, we can draw the following conclusions:

- LLAMA3.2-1B and LLAMA3.2-3B show no significant performance differences across almost all datasets, indicating similar multilingual capabilities. The performance differentiation of small-size models below 7B is slightly worse.

Dataset	MISTRAL	LLAMA3.2	LLAMA3.1	QWEN2.5	GEMMA2	>70B	7B-14B	<7B
FLORES-200	2/2	2/2	1/2	4/7	3/3	3/3	2/5	3/6
MHELLASWAG	2/2	1/2	2/2	6/7	2/3	2/3	5/5	5/6
XNLI	2/2	1/2	2/2	5/7	3/3	2/3	3/5	5/6
HUMANEVAL-XL (Python)	2/2	1/2	2/2	2/7	1/3	3/3	3/5	3/6
HUMANEVAL-XL (JavaScript)	2/2	1/2	2/2	5/7	3/3	2/3	5/5	5/6
HUMANEVAL-XL (Java)	2/2	1/2	2/2	4/7	3/3	2/3	3/5	3/6
MGSM	2/2	1/2	2/2	6/7	3/3	1/3	4/5	4/6
MLOGIQA	2/2	1/2	2/2	6/7	3/3	2/3	3/5	3/6
MIFEVAL	2/2	1/2	2/2	6/7	2/3	3/3	2/5	4/6

Table 4: All tested models are categorized into 8 categories based on model size and series. This table presents the utility of each dataset in distinguishing the performances of paired models within the same category. A value closer to 1 indicates higher utility for the dataset, with a value of 1 signifying that all models demonstrate distinguishable performances. Conversely, a numerator of 1 indicates that no models are distinguishable on that dataset. We set the threshold at 0.5, where each value is considered effective or ineffective in distinguishing the performances of models with the specified dataset.

- Compared to JavaScript and Java, most models show poor performance differentiation in Python. According to the Appendix C, the average score of all the tested open-source models in Python is 90.46, significantly higher than the scores in the other two languages (48.95 and 46.66, respectively), indicating that all models have a strong knowledge grasp in Python.
- All selected datasets can distinguish between models in the majority of categories, which verifies the effectiveness of all datasets included in P-MMEVAL.

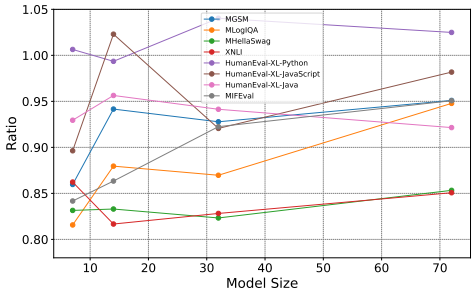

Figure 1: Illustration on the ratio of non-English performance to English performance with increasing model sizes of QWEN2.5.

6.2 Performances on English vs. Non-English Benchmarks

To preliminarily explore the relationship between non-English ability and English ability of the model, we use various sizes of the QWEN2.5 model (7B, 14B, 32B, and 72B) to evaluate their performance on six datasets with parallel samples in different languages. For each dataset, we calculate the ratio of the average score achieved on the test sets in all nine non-English languages to the score achieved on the test data in English. We do not consider models smaller than 7B, as these models are easily influenced by prompts, leading to performance fluctuations.

Figure 1 illustrates the trend of the ratio of non-English performance to English performance as model sizes increase. On five datasets, the model's non-English performance appears limited by its English performance. However, on the three programming languages (Python, JavaScript, Java) of HUMANEVAL-XL dataset, the models achieve comparable performance in both English and non-English test sets. This means that code knowledge is less dependent on natural language. When the model size increases, we observe that: 1) As for instruction-following ability, the gap between non-English data and English data is narrowing. 2) The ratio of capability-specialized datasets outperforms those of fundamental understanding datasets.

7 Conclusion

In this paper, we first present a pipeline for benchmark selection, which guides the finding and selecting of effective benchmarks for quantifying the multilingual performances of LLMs. Then, we introduce a comprehensive multilingual multitask benchmark, P-MMEVAL, which evaluates LLMs across both fundamental and capability-specialized tasks, ensuring consistent language coverage and providing parallel samples in multiple languages. Furthermore, we conduct extensive experiments on representative multilingual model series. These findings provide valuable guidance for future research, highlighting the importance of balanced and comprehensive training data, effective prompt engineering, and the need for targeted improvements in specific language capabilities.

Limitations

Through the above experiments and analyses, we summarize the following limitations:

1) Language Coverage: While P-MMEval currently covers 10 languages from 8 language families, there is a need to include more languages to better represent global linguistic diversity. Future work will focus on expanding the language coverage to ensure a more comprehensive evaluation of multilingual LLMs.

2) Task Diversity: P-MMEval includes 7 representative tasks, but the rapidly evolving field of LLMs demands a broader range of tasks. Future work will focus on expanding the benchmark to cover more diverse and challenging tasks, providing a more thorough assessment of multilingual LLMs.

Ethics Statement

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Uttama Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4232–4267. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Akari Asai, Sneha Kudugunta, Xinyan Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2024. BUFFET: benchmarking large language models for few-shot cross-lingual transfer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1771–1800. Association for Computational Linguistics.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7421–7454. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:abs/2309.16609*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:abs/2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:abs/2207.04672*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv e-prints*, pages arXiv–2307.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:abs/2407.21783*.

Andy Field. 2005. Discovering statistics using ibm spss statistics. *Sage*.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George F. Foster, Alon Lavie, and Ondrej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 733–774. Association for Computational Linguistics.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2024. Are we done with mmlu? *arXiv preprint arXiv:abs/2406.04127*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Samin Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguis-*

tics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4693–4703. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:abs/2003.11080*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017a. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017b. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7315–7330. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Bruce Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:abs/2004.01401*, abs/2004.01401.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Qiwei Peng, Yekun Chai, and Xuhong Li. 2024. Humaneval-xl: A multilingual code generation benchmark for cross-lingual natural language generalization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 8383–8394. ELRA and ICCL.

Edoardo Maria Ponti, Goran Glavas, Olga Majewska, Qianchu Liu, Ivan Vulic, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2362–2376. Association for Computational Linguistics.

Alessandro Raganato, Tommaso Pasini, José Camacho-Collados, and Mohammad Taher Pilehvar. 2020. Xl-wic: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7193–7206. Association for Computational Linguistics.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila

Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:abs/2408.00118*.

Sebastian Ruder, Noah Constant, Jan A. Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10215–10245. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Alexey Tikhonov and Max Ryabinin. 2021. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3534–3546. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:abs/2407.10671*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3685–3690. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:abs/2311.07911*.

Dataset	Native	EN	EN-Few-shot
MMMLU	44.30	44.69	45.70
MLogiQA	42.27	41.96	44.88
MGSM	62.13	64.17	63.28
MHellaSwag	52.03	53.37	59.07
XNLI	54.49	55.31	64.08
Flores-200	30.00	24.31	29.18

Table 5: Comparison on P-MMEval using three different prompt settings.

A The Impact of Different Prompts on Model Performance

We explore three different prompting strategies: EN, Native, and En-Few-Shot. Table 5 illustrates the average performance of all evaluated open-source models on various datasets of P-MMEval. Overall, the performance difference between the EN prompt and the Native prompt is minimal, remaining within 2%, indicating no substantial performance gap. However, in the case of the Flores-200, the EN prompt results in a marked decline in performance compared to the Native prompt. We observe that models always generate responses in English when English instructions are used to describe the task for non-English data for generation tasks. On various datasets, the few-shot prompt leads to better model performance than the zero-shot prompt, as models achieve a higher success rate in extracting answers in the few-shot setting.

B Expert Translation Review Results on Each Dataset

To supplement the missing multilingual portions in each dataset, a strategy that combines machine translation with professional human review is adopted. Table 6 shows the percentage of modifications made by professional translators to the machine translation results generated by GPT-4o. The main types of translation errors include omissions, incorrect translation order, and improper use of localized vocabulary.

C Evaluation Results on Three Programming Languages of HumanEval-XL

Table 7 shows the evaluation results of all tested models on three programming languages of HumanEval-XL. Model performance in Python greatly exceeds the performance in the other two programming languages. For instance, Gemma2-2B scores 98.13 in Python, compared to 29.25 in JavaScript and 27.25 in Java. Additionally, as the model size increases, there is a noticeable improvement in performance for both JavaScript and Java.

D Model performance on each language with Increasing Model Sizes

This section analyzes the trend of the performance of the model in each language with increasing model sizes. We only report the average performance on four capability-specialized datasets (HumanEval-XL, MGSM, MLogiQA, and MIFEval). In addition, we do not consider models smaller than 7B, as these models are easily influenced by prompts, leading to performance fluctuations. Model performance varies by language, with English demonstrating the strongest capabilities, while Thai and Japanese show the weakest.

E Dataset Utility

To quantify the utility of each dataset, we employ paired-sample T-tests for each pair of models within the same categories. Inspired by (Freitag et al., 2021), our main motivation is to try to divide models in the same category into several groups based on their pairwise significance gaps, where all model pairs in the same group do not have significant performance gaps, and performances of all model pairs from different groups are hard to be fully distinguished. Given the list of all models $\mathbf{m} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_m]$, we recurrently gather some of the models into the same group $\mathbf{\Omega}_i = \{\mathbf{m}_{\pi_1}, \mathbf{m}_{\pi_2}, \cdots, \mathbf{m}_{\pi_k}\}, \pi_j \in [1, 2, \cdots, m]$ for $j \in [1, 2, \cdots, k]$ at the i-th step, where: 1) for each model $\mathbf{m}_{\pi_j}$ in $\mathbf{\Omega}_i$, it does not have a significant performance gap against any

Dataset	*zh*	*ar*	*es*	*ja*	*ko*	*th*	*fr*	*pt*	*vi*
XNLI	/	/	/	22.50	11.67	/	/	10.83	/
MHELLASWAG	/	/	/	82.50	77.50	26.67	/	/	/
HUMANEVAL-XL	/	/	/	42.50	23.75	31.25	/	/	/
MGSM	/	9.20	/	/	32.80	/	/	5.60	27.20
MLOGIQA	/	22.50	30.00	51.25	33.75	46.25	3.75	46.25	18.75
MMMLU	/	/	/	/	/	26.00	13.50	/	/
MIFEVAL	25.50	23.81	20.00	45.71	36.19	37.14	21.90	17.14	24.76

Table 6: The table presents the percentage of modifications made by professional translators to the machine translation results.

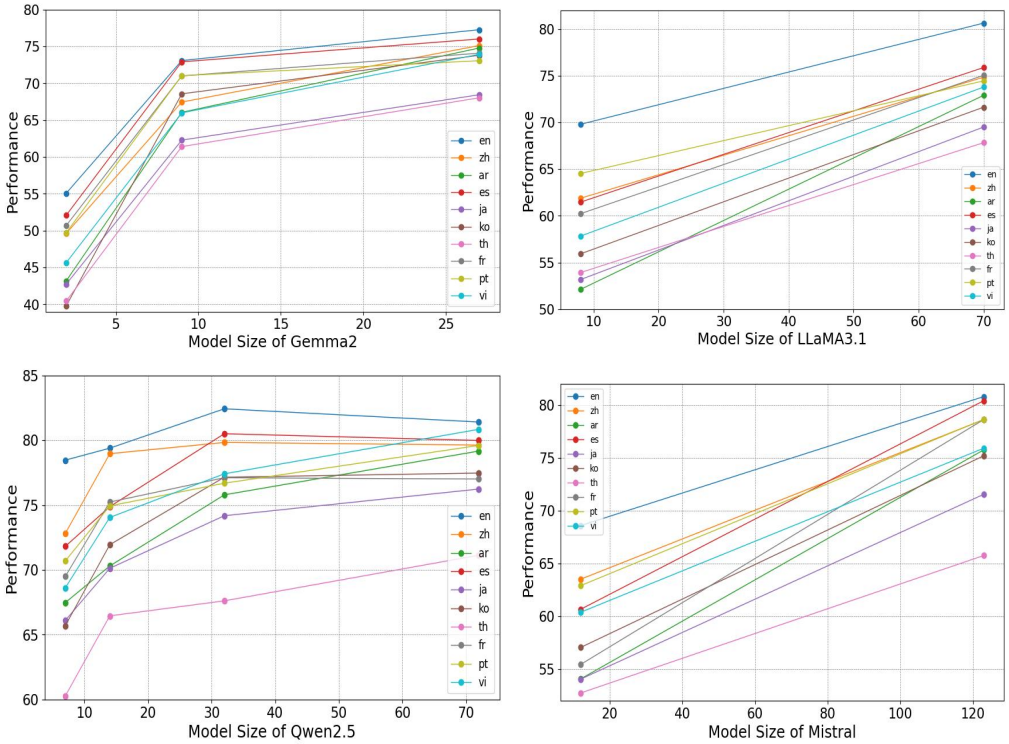

Figure 2: This figure illustrates the trend of the performance of the model in each language with increasing model sizes.

model in $\mathbf{\Omega}_i$ except itself:

$$f_1 = \begin{cases} \text{true if } \mathcal{T}(\mathbf{m}_{\pi_j}, \mathbf{m}_{\pi_p}) > \theta \text{ holds for any} \\ \qquad\qquad p \in [1, 2, \cdots, k], j \neq p; \\ \text{false otherwise;} \end{cases} \tag{1}$$

2) for each model in $\mathbf{\Omega}_i$, it has significant performance gaps against all the model not in $\mathbf{\Omega}_i$:

$$f_2 = \begin{cases} \text{true if } \mathcal{T}(\mathbf{m}_{\pi_j}, \mathbf{m}_{p}) < \theta \text{ holds for all} \\ \qquad\qquad p \notin [\pi_1, \pi_2, \cdots, \pi_k]; \\ \text{false otherwise;} \end{cases} \tag{2}$$

where $\mathcal{T}(\cdot, \cdot)$ returns the p-value of the performances between two given models, and θ represents the threshold for denoting significance level. The group $\mathbf{\Omega}_i$ is fixed if f_1 and f_2 both hold true. Such a recurrent process continues till each model is gathered into one specific group.[8]

After gathering all models into several groups, we use the ratio of the number of such groups to the number of models to describe the utility of the specific dataset. A higher ratio means that we have more gathered groups, indicating that the benchmark is of high utility in distinguishing the performances of models. On the contrary, a lower ratio

[8] See Algorithm 1 in Appendix E for more details.

	Python	JavaScript	Java
LLaMA3.2-1B	92.13	9.38	11.63
LLaMA3.2-3B	91.50	9.75	11.00
Qwen2.5-0.5B	78.38	14.25	9.13
Qwen2.5-1.5B	81.63	35.88	28.25
Qwen2.5-3B	84.00	53.75	44.50
Gemma2-2B	98.13	29.25	27.25
LLaMA3.1-8B	96.38	46.88	66.63
Qwen2.5-7B	86.75	68.00	60.88
Gemma2-9B	98.75	54.63	56.50
Mistral-Nemo	93.25	39.63	39.25
Qwen2.5-14B	84.50	72.75	61.25
Qwen2.5-32B	89.38	73.13	65.13
Gemma2-27B	99.63	63.75	66.63
LLaMA3.1-70B	98.75	63.38	62.13
Qwen2.5-72B	85.63	75.00	67.38
Mistral-Large	88.63	73.88	69.00
GPT-4o	89.13	77.88	64.13
Claude-3.5-sonnet	99.75	74.00	75.00

Table 7: The table presents the performance on three programming languages of HumanEval-XL.

means that most of the models can be gathered into the same group, denoting that the benchmark may hardly tell which model performs better than any other model.

The algorithm for quantifying the utility of each benchmark dataset is presented in Algorithm 1.

F Significance Detection on Each Dataset

The section illustrates the significant difference between models' pairwise performance for all categories of models.

G The Prompt Utilized for Each Dataset

The section presents the inference prompt utilized for each dataset.

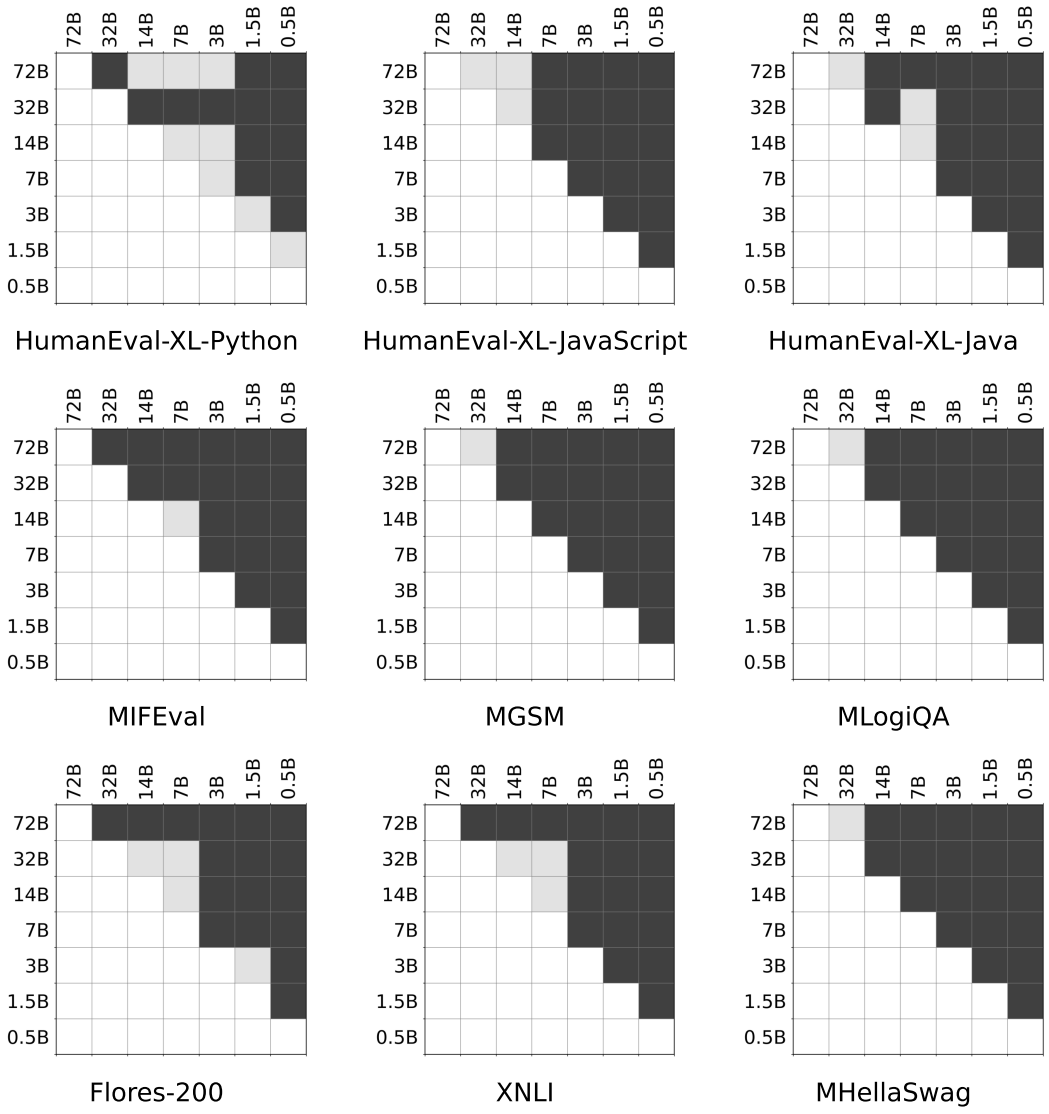

Figure 3: This figure illustrates the significant difference in pairwise performance among QWEN2.5 series models. Black blocks indicate that the p-values of paired t-tests between the corresponding models (vertical and horizontal) are less than 0.01, while gray blocks indicate p-values greater than 0.01.

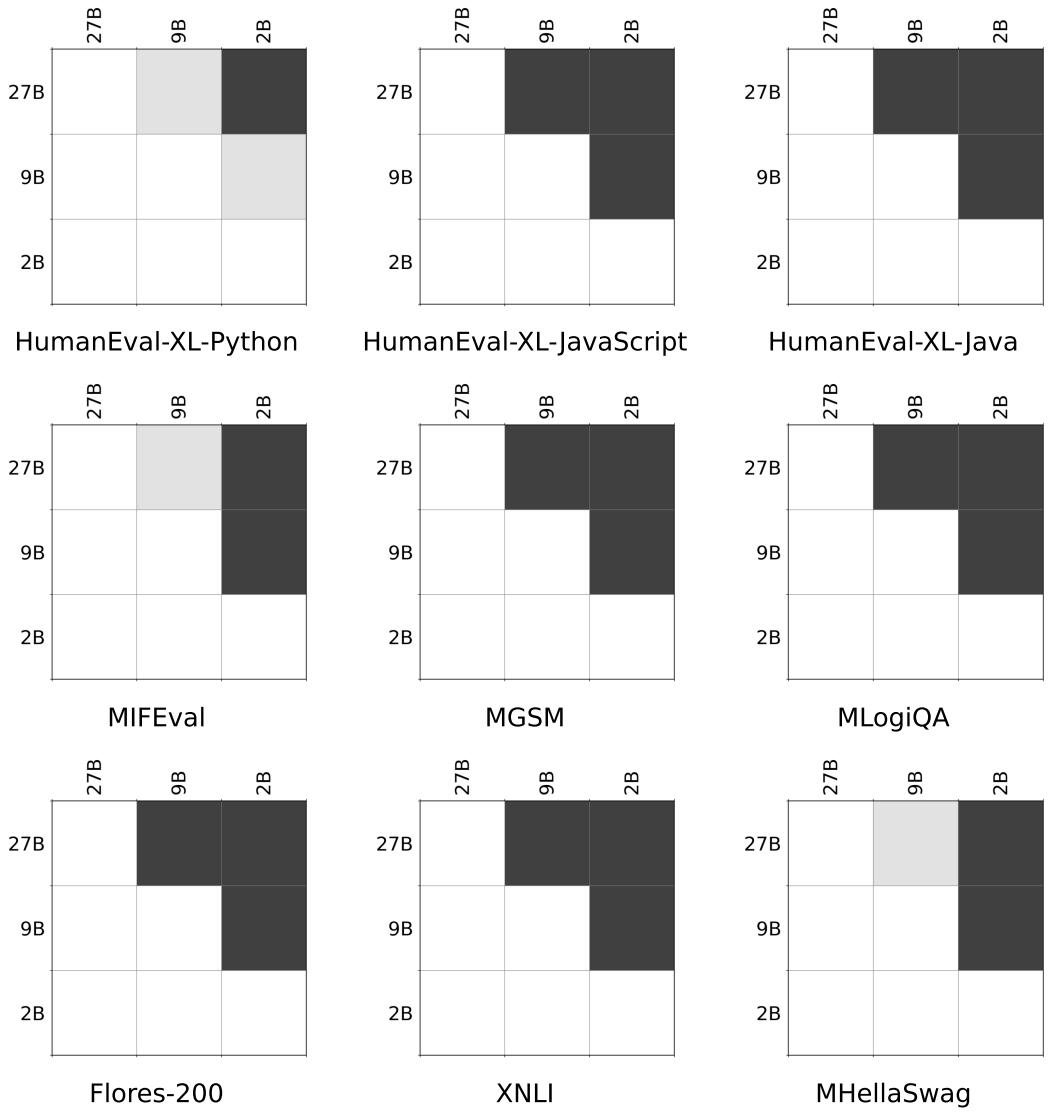

Figure 4: This figure illustrates the significant difference in pairwise performance among GEMMA2 series models.

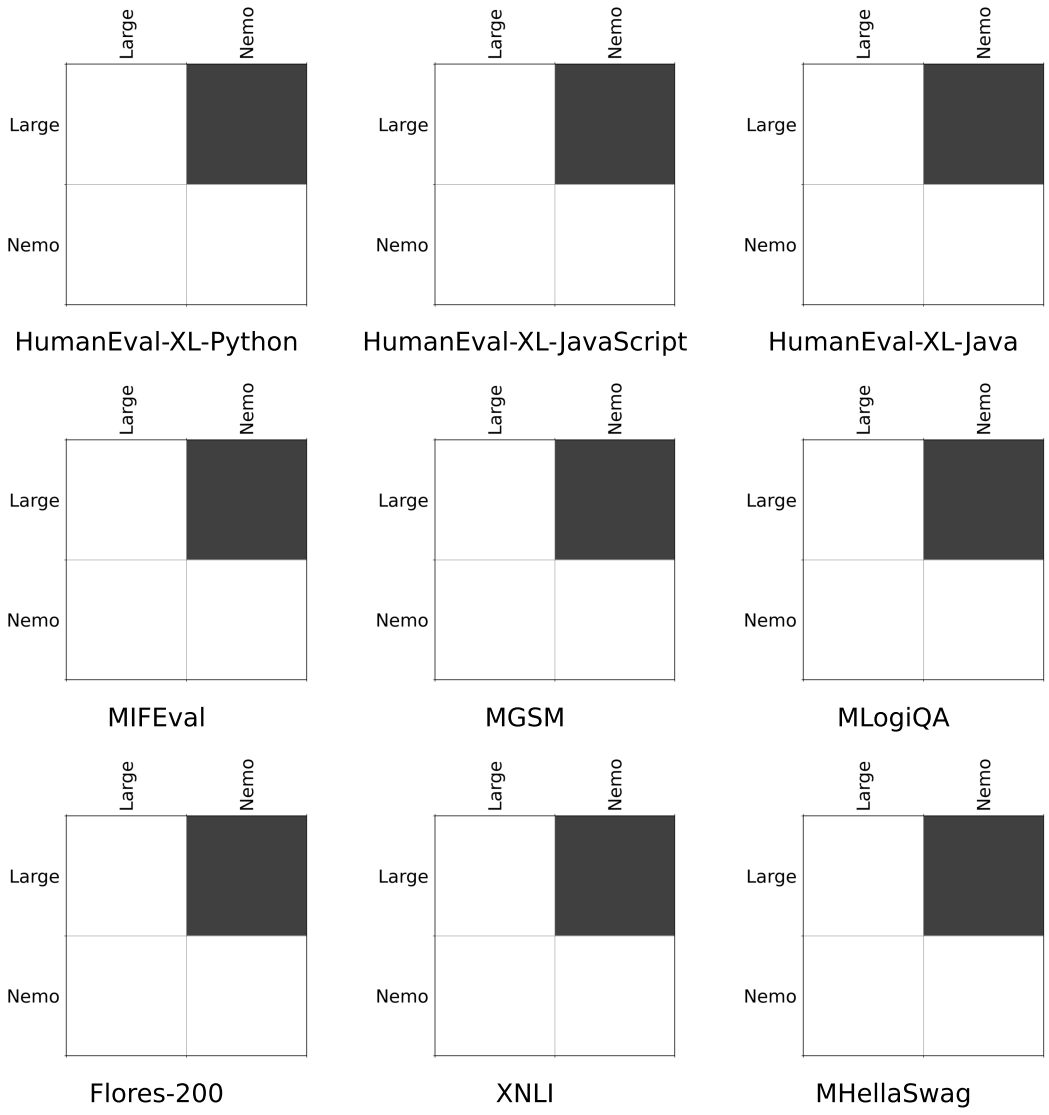

Figure 5: This figure illustrates the significant difference in pairwise performance among MISTRAL series models.

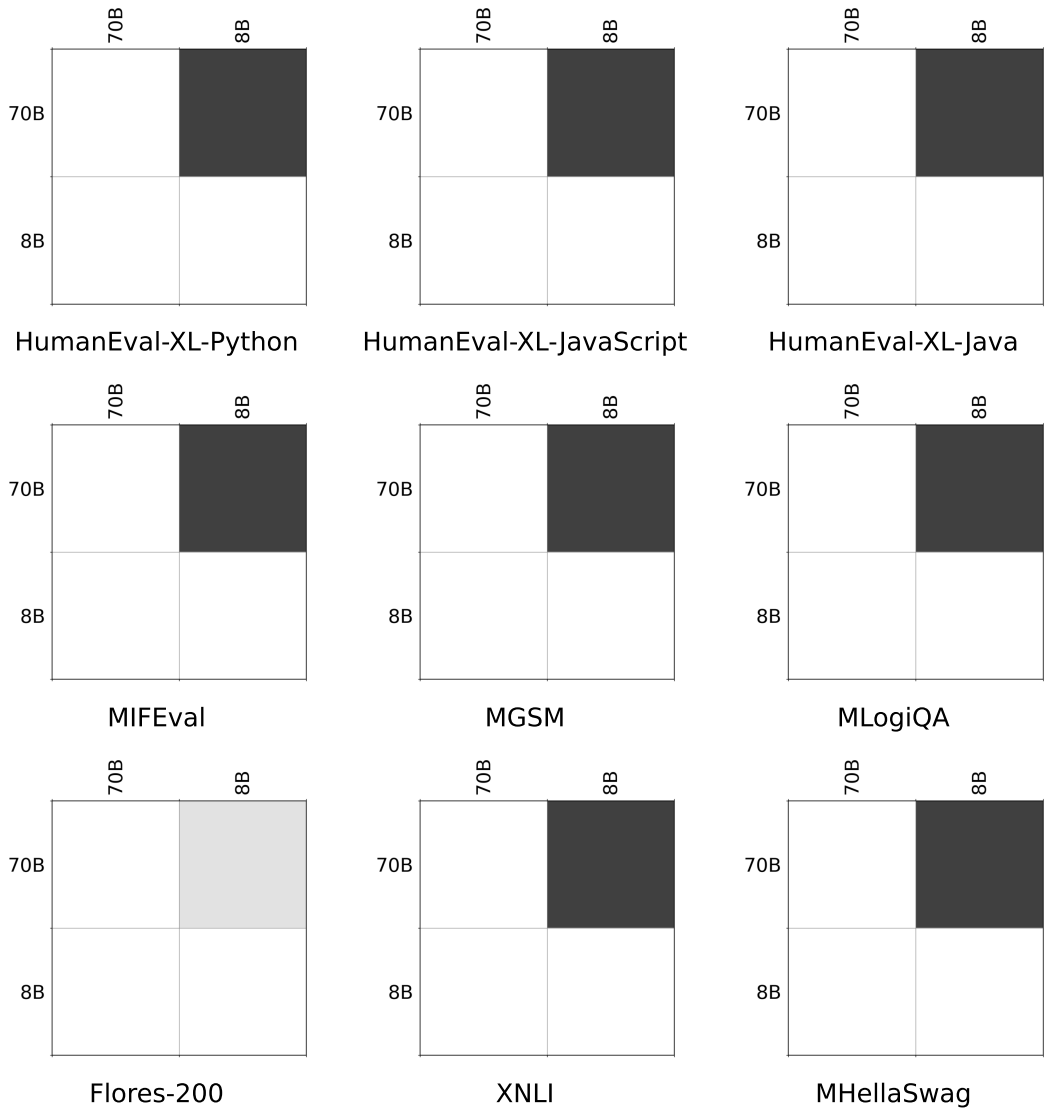

Figure 6: This figure illustrates the significant difference in pairwise performance among LLAMA3.1 series models.

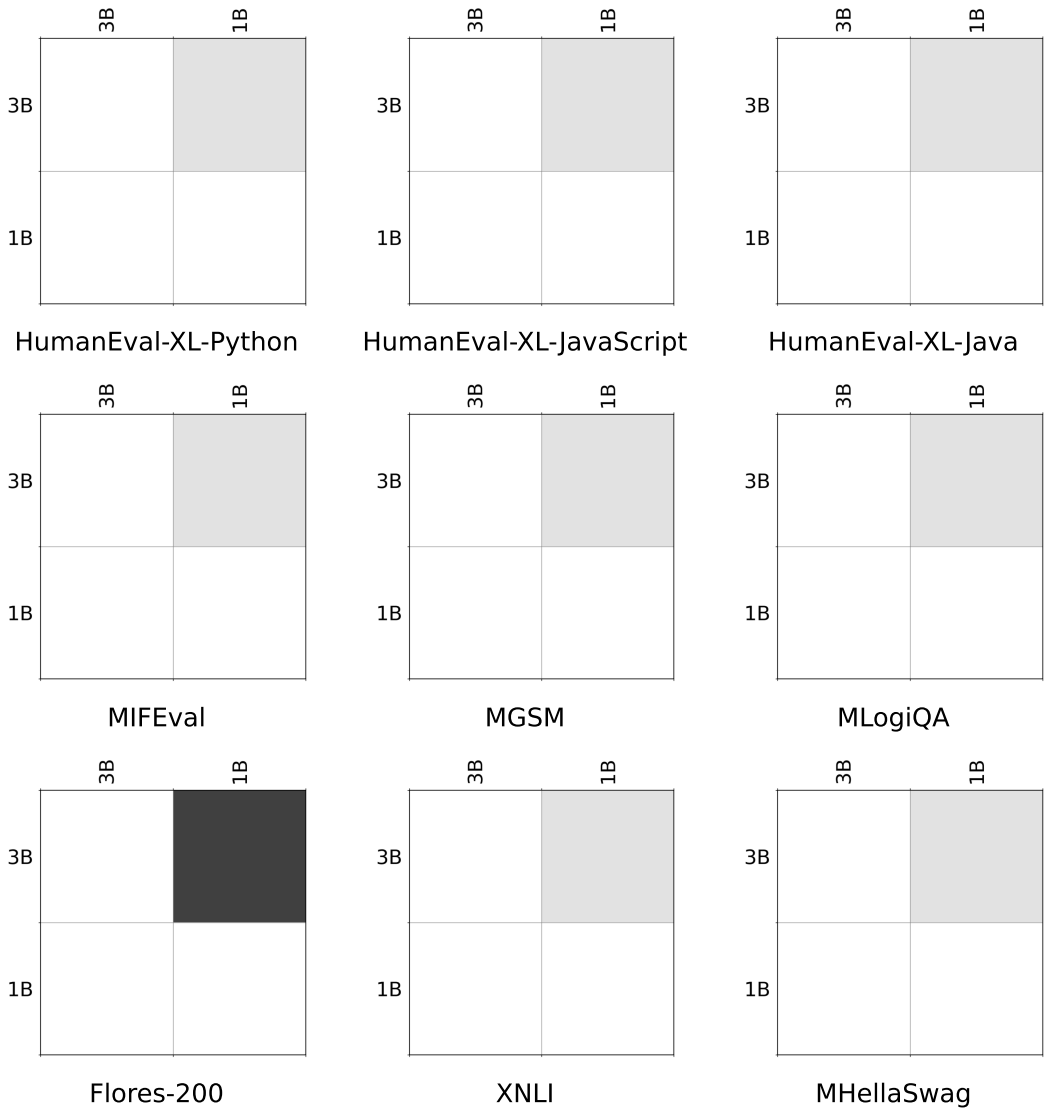

Figure 7: This figure illustrates the significant difference in pairwise performance among LLAMA3.2 series models.

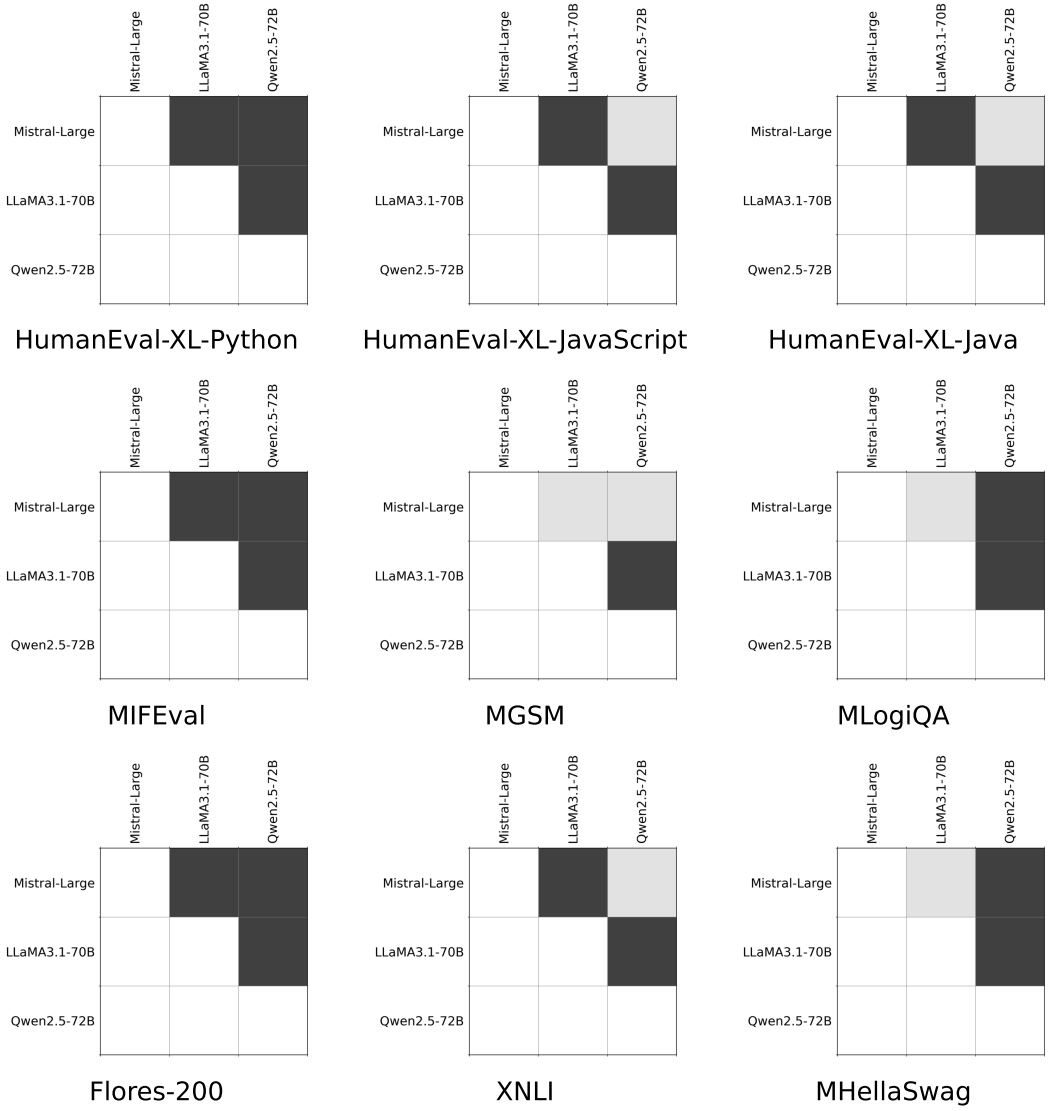

Figure 8: This figure illustrates the significant difference in pairwise performance among models with more than 70 billion parameters.

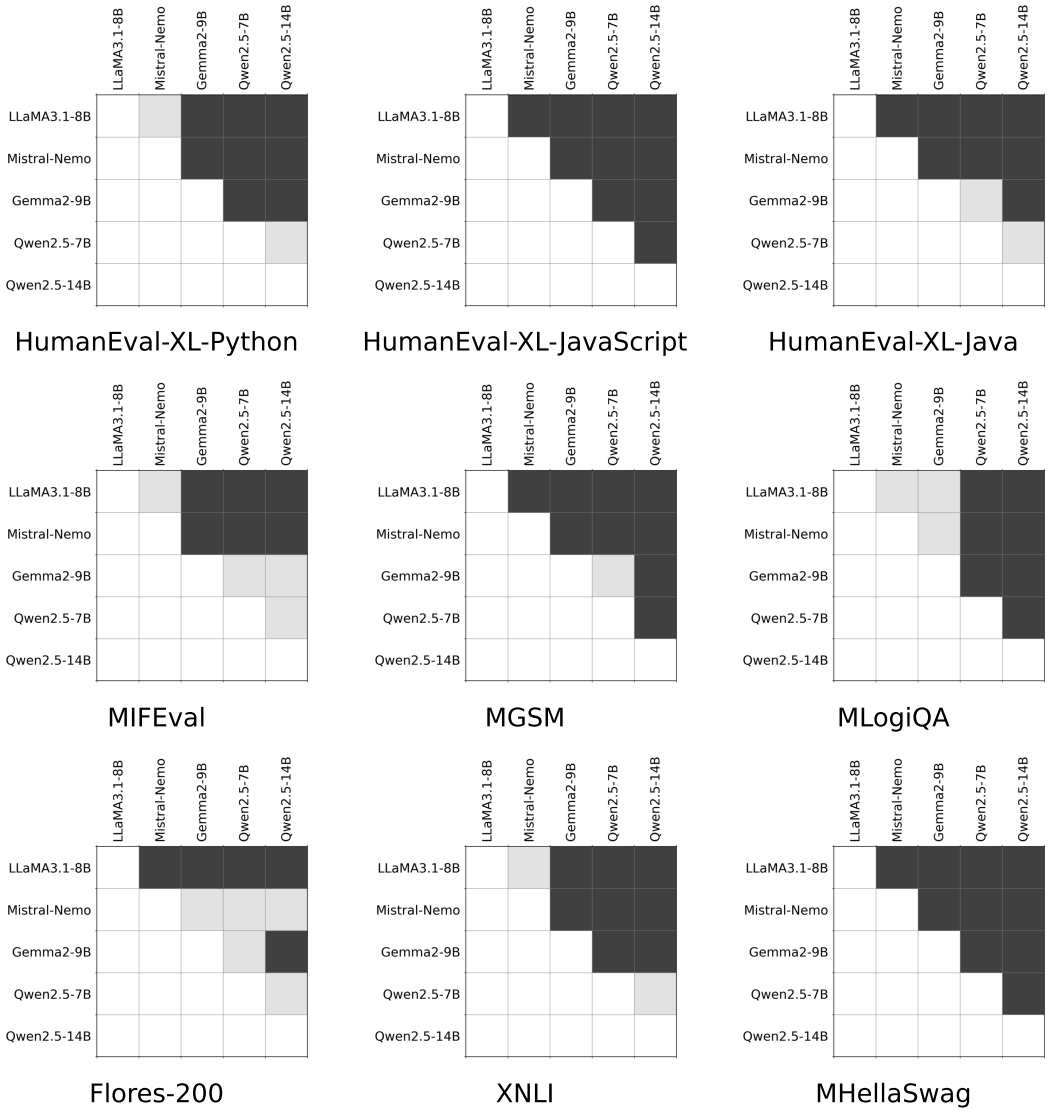

Figure 9: This figure illustrates the significant difference in pairwise performance among models with 7 to 14 billion parameters.

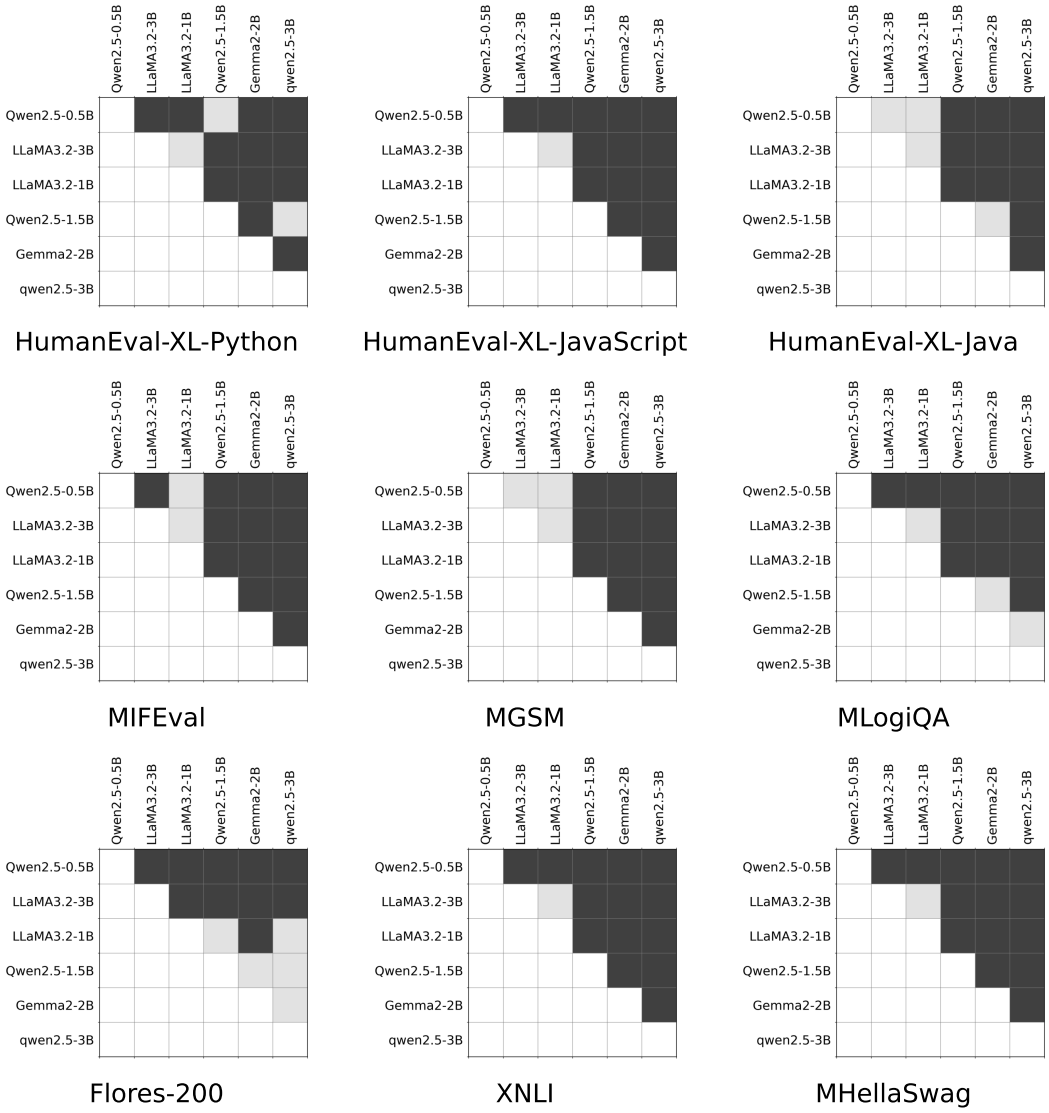

Figure 10: This figure illustrates the significant difference in pairwise performance among models with fewer than 7 billion parameters.

Algorithm 1 Algorithm for Quantifying the Utility of a Specific Benchmark Dataset

Input: Model ids $\mathbf{m} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_m]$, paired-sample T-test p-values among all pairs of models $p_{\mathbf{m}_i, \mathbf{m}_j} \in \mathbb{R}(0 \leq i, j \leq m, p_{ij} = p_{ji}, i \neq j)$, significance threshold $\theta \in \mathbb{R}$
Output: The number of sets $|\mathbf{\Omega}|$, where $\mathbf{\Omega}$ is a list of sets $\mathbf{\Omega} = [\mathbf{\Omega}_1, \mathbf{\Omega}_2, \cdots, \mathbf{\Omega}_s]$, and each set contains several models $\mathbf{\Omega}_i = \{\mathbf{m}_{\pi_1}, \mathbf{m}_{\pi_2}, \cdots, \mathbf{m}_{\pi_k}\}, \mathbf{\Omega}_i \neq \phi, |\mathbf{\Omega}| = k \leq m, \pi_j \in [1, 2, \cdots, m]$ for $j \in [1, 2, \cdots, k]$

$\mathbf{\Omega} \leftarrow []$ ▷ Initialize with an empty list
$\mathbf{z} = \mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_m$
while $\mathbf{z} \neq \phi$ **do**
    $\mathbf{x} \leftarrow \{\mathbf{z}_1\}$ ▷ Initialize the current set with the first model id
    $\mathbf{y} \leftarrow \mathbf{z} - \mathbf{x}$
    while $\mathbf{y} \neq \phi$ **do**
        Initialize Γ as a matrix full of ϕ
        for $c \in \mathbf{x}$ **do**
            for $d \in \mathbf{y}$ **do**
                if $p_{c,d} < \theta$ **then**
                    $\Gamma[c, d] \leftarrow$ true
                    $\Gamma[d, c] \leftarrow$ true ▷ The gap is significant
                else
                    $\Gamma[c, d] \leftarrow$ false
                    $\Gamma[d, c] \leftarrow$ false ▷ The gap is not significant
        if $\Gamma[c, d] =$ false for any $c \in \mathbf{x}, d \in \mathbf{y}$ **then** ▷ Some paired models do not have significant performance gaps
            for $d \in \mathbf{y}$ **do**
                if $\Gamma[c, d] =$ false for any $c \in \mathbf{x}$ **then**
                    $\mathbf{x} \leftarrow \mathbf{x} + \{d\}$
                    $\mathbf{y} \leftarrow \mathbf{y} - \{d\}$ ▷ Moving model d into the same group
        else ▷ Each model from $\mathbf{x}$ has significant gap against each model from $\mathbf{y}$
            $\mathbf{\Omega} \leftarrow \mathbf{\Omega} + [\mathbf{x}]$ ▷ Appending the new group $\mathbf{x}$ into $\mathbf{\Omega}$
            $\mathbf{z} \leftarrow \mathbf{z} - \mathbf{x}$ ▷ Removing the processed model ids from $\mathbf{z}$
return $|\mathbf{\Omega}|$ ▷ Return the number of groups

EN prompt for FLORES-200-en-x:
All: "Translate this sentence from English to {tgt_lang}.\n\n{src}\n"

Native prompt for FLORES-200-en-x:
zh: "将这个句子从英语翻译成中文。\n\n{src}"

th: "แปลประโยคนี้จากภาษาอังกฤษเป็นภาษาไทย.\n\n{src}"

ar: "ترجم هذه الجملة من الإنجليزية إلى العربية.\n\n{src}"

es: "Traduce esta oración del inglés al español.\n\n{src}"

ja: "この文を英語から日本語に翻訳してください。\n\n{src}"

ko: "이 문장을 영어에서 한국어로 번역하세요.\n\n{src}"

fr: "Traduisez cette phrase de l'anglais en français.\n\n{src}"

pt: "Traduza esta frase do inglês para o português.\n\n{src}"

vi: "Dịch câu này từ tiếng Anh sang tiếng Việt.\n\n{src}"

EN prompt for FLORES-x-en:
All: "Translate this sentence from {src_lang} to English.\n\n{src}\n"

Figure 11: This figure presents the prompt for the Flores-200 dataset.

EN prompt for MHELLASWAG:
All: "Input: {premise}\nOptions: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nPick the correct ending for the sentence from A, B, C, and D, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C or D."

Native prompt for MHELLASWAG:
zh: "输入：{premise}\n选项：\nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\n从 A, B, C 或者 D 中选出正确的句子结尾，并按照以下 JSON 格式返回：\n{'answer': '[choice]'}\n其中 [choice] 必须是 A, B, C 或者 D 其中之一。"

en: "Input: {premise}\nOptions: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nPick the correct ending for the sentence from A, B, C, and D, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C or D."

vi: "Nhập: {premise}\nLựa chọn: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nChọn kết thúc đúng cho câu từ A, B, C và D, và trả về theo định dạng JSON sau:\n{'answer': '[choice]'}\nTrong đó [choice] phải là một trong các A, B, C hoặc D."

th: "ข้อมูลนำเข้า: {premise}\nตัวเลือก: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nเลือกตอนจบที่ถูกต้องสำหรับประโยคจา A, B, C และ D แล้วส่งคืนในรูปแบบ JSON ดังต่อไปนี้:\n{'answer': '[choice]'}\nโดย [choice] จะต้องเป็นหนึ่งใน A, B, C หรือ D."

ar: "الإدخال: {premise}\nالخيارات: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nاختر النهاية الصحيحة للجملة من A و B و C و D، وأعد كتابة الإجابة بتنسيق JSON التالي:\n{'answer': '[choice]'}\nحيث يجب أن يكون [choice] واحدًا من A أو B أو C أو D."

es: "Entrada: {premise}\nOpciones: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nElija el final correcto para la oración de A, B, C y D, y devuélvalo en el siguiente formato JSON:\n{'answer': '[choice]'}\ndonde [choice] debe ser uno de A, B, C o D."

ja: "入力: {premise}\n選択肢: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nA、B、C、Dから文の正しい結末を選び、次のJSON形式で返してください：\n{'answer': '[choice]'}\nここで、[choice]はA、B、C、またはDのいずれかでなければなりません。"

ko: "입력: {premise}\n옵션: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nA, B, C, D 중에서 문장의 올바른 엔딩을 선택하고, 다음 JSON 형식으로 반환하십시오:\n{'answer': '[choice]'}\n여기서 [choice]는 A, B, C 또는 D 중 하나여야 합니다."

fr: "Entrée : {premise}\nOptions : \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nChoisissez la fin correcte de la phrase parmi A, B, C et D, et renvoyez-la dans le format JSON suivant :\n{'answer': '[choice]'}\noù [choice] doit être l'un de A, B, C ou D."

pt: "Entrada: {premise}\nOpções: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nEscolha o final correto para a frase de A, B, C e D, e retorne-o no seguinte formato JSON:\n{'answer': '[choice]'}\nonde [choice] deve ser uma das opções A, B, C ou D."

Figure 12: This figure presents the prompt for the MHellaSwag dataset.

EN prompt for XNLI:
All: "Take the following as truth: {premise}\nThen the following statement: "{hypothesis}" is\nOptions: \nA. true\nB. inconclusive\nC. false\nSelect the correct option from A, B, and C, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, and C."

Native prompt for XNLI:
zh: "假设以下内容为真：{premise}\n考虑以下陈述："{hypothesis}"\n该陈述是：\n选项：\nA. 真实的\nB. 无法确定\nC. 虚假的\n从 A, B 或者 C 中选择正确的选项，并按以下JSON格式返回：\n{'answer': '[choice]'}\n其中 [choice] 必须是 A, B 或者 C 其中之一。"

en: "Take the following as truth: {premise}\nThen the following statement: "{hypothesis}" is\nOptions: \nA. true\nB. inconclusive\nC. false\nSelect the correct option from A, B, and C, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, and C."

th: "ให้ถือว่าเป็นความจริง: {premise}\nแล้วข้อความต่อไปนี้: "{hypothesis}" เป็น\nตัวเลือก: \nA. จริง\nB. ไม่แน่นอน\nC. เท็จ\nเลือกตัวเลือกที่ถูกต้องจาก A, B, และ C และส่งคืนในรูปแบบ JSON ดังต่อไปนี้:\n{'answer': '[choice]'}\nโดยที่ [choice] ต้องเป็นหนึ่งใน A, B, และ C."

ar: "اعتبر ما يلي كحقيقة: {premise}\nثم الجملة التالية: "{hypothesis}" هي\nالخيارات: \nA. صحيحة\nB. غير ذلك\nC. لا يمكن تحديده\nاختر الخيار الصحيح من A وB وC في تنسيق JSON التالي:\n{'answer': '[choice]'}\n[choice] يجب أن يكون أحد A أو B أو C."

es: "Tome lo siguiente como verdad: {premise}\nEntonces la siguiente afirmación: "{hypothesis}" es\nOpciones: \nA. verdadera\nB. inconclusa\nC. falsa\nSeleccione la opción correcta de A, B y C, y devuélvala en el siguiente formato JSON:\n{'answer': '[choice]'}\ndonde [choice] debe ser una de A, B y C."

ja: "次の内容を真実とみなしてください：{premise}\n次の文："{hypothesis}" は\n選択肢：\nA. 真\nB. 不確定\nC. 偽\nA、B、Cの中から正しい選択肢を選び、次のJSON形式で返してください：\n{'answer': '[choice]'}\nここで、[choice]はA、B、Cのいずれかでなければなりません。"

ko: "다음 내용을 진실로 간주하십시오: {premise}\n그렇다면 다음 진술: "{hypothesis}"는\n옵션: \nA. 사실\nB. 결론을 내릴 수 없음\nC. 거짓\nA, B, C 중에서 올바른 옵션을 선택하고 다음 JSON 형식으로 반환하십시오:\n{'answer': '[choice]'}\n여기서 [choice]는 A, B 및 C 중 하나여야 합니다."

fr: "Prenez ce qui suit comme vérité : {premise}\nAlors, l'affirmation suivante : "{hypothesis}" est\nOptions : \nA. vraie\nB. inconclusive\nC. fausse\nSélectionnez l'option correcte parmi A, B et C, puis renvoyez-la dans le format JSON suivant :\n{'answer': '[choice]'}\noù [choice] doit être l'un de A, B et C."

pt: "Considere o seguinte como verdade: {premise}\nEntão, a seguinte afirmação: "{hypothesis}" é\nOpções: \nA. verdadeira\nB. inconclusiva\nC. falsa\nSelecione a opção correta de A, B e C e retorne-a no seguinte formato JSON:\n{'answer': '[choice]'}\nonde [choice] deve ser uma das opções A, B ou C."

vi: "Xem điều sau đây là đúng: {premise}\nVậy tuyên bố sau đây: "{hypothesis}" là\nCác lựa chọn: \nA. đúng\nB. không kết luận\nC. sai\nChọn lựa chọn đúng từ A, B và C, và trả lại nó theo định dạng JSON sau:\n{'answer': '[choice]'}\ntrong đó [choice] phải là một trong A, B và C."

Figure 13: This figure presents the prompt for the XNLI dataset.

Native prompt for MGSM:

en: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "The answer is ". Do not add anything other than the integer answer after "The answer is ".\n\n{question}"

es: "Resuelve este problema matemático. Proporciona los pasos de razonamiento antes de dar la respuesta final en la última línea por sí misma en el formato de "La respuesta es ". No añadas nada más que la respuesta entera después de "La respuesta es ".\n\n{question}"

fr: "Résolvez ce problème de mathématiques. Donnez les étapes de raisonnement avant de fournir la réponse finale sur la dernière ligne elle-même dans le format de "La réponse est ". N'ajoutez rien d'autre que la réponse entière après "La réponse est ".\n\n{question}"

ja: "の数学の問題を解いてください。最終的な答えを出す前に、解答の推論過程を記述してください。そして最後の行には "答えは " の形式で答えを記述し、その後には整数の答え以外何も追加しないでください。\n\n{question}"

th: "แก้ปัญหาคณิตศาสตร์นี้ ให้ให้ขั้นตอนการใช้เหตุผลก่อนที่จะให้คำตอบสุดท้ายในบรรทัดสุดท้ายโดยอยู่ในรูปแบบ "คำตอบคื " ไม่ควรเพิ่มอะไรนอกจากคำตอบที่เป็นจำนวนเต็มหลังจ "คำตอบคื "\n\n{question}"

zh: "解决这个数学问题。在最后一行给出答案前，请提供推理步骤。最后一行应该以 "答案是 " 的形式独立给出答案。在 "答案是 " 后不要添加除整数答案之外的任何内容。\n\n{question}"

ar: "قم بحل هذه المسألة الرياضية. قبل إعطاء الإجابة في السطر الأخير، يجري تقديم خطوات الحل. يجب أن يتم تقديم الإجابة في السطر الأخير بشكل مستقل على شكل "الجواب هو ". لا تُضِف أي شيء بعد "الجواب هو " سوى العدد الصحيح للإجابة.\n\n{question}"

ko: "이 수학 문제를 해결하십시오. 마지막 줄에 답을 제시하기 전에 추론 단계를 제공하십시오. 마지막 줄은 "답변은 " 형식으로 독립적으로 답을 제시해야 합니다. "답변은 " 뒤에는 정수답 이외의 어떤 것도 추가하지 마십시오.\n\n{question}"

pt: "Resolva este problema matemático. Antes de dar a resposta na última linha, por favor, forneça os passos de raciocínio. A última linha deve apresentar a resposta de forma independente, começando com "A resposta é ". Após "A resposta é " não adicione nada além da resposta em número inteiro.\n\n{question}"

vi: "Giải quyết vấn đề toán học này. Trước khi đưa ra đáp án ở dòng cuối cùng, hãy cung cấp các bước lập luận. Dòng cuối cùng nên đưa ra đáp án dưới dạng "Câu trả lời là " một cách độc lập. Không thêm bất cứ nội dung nào ngoài đáp án là số nguyên sau "Câu trả lời là ".\n\n{question}"

Figure 14: This figure presents the Native prompt for the MGSM dataset.

EN prompt for MGSM:

en: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "The answer is ". Do not add anything other than the integer answer after "The answer is ".\n\n{question}"

es: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "La respuesta es ". Do not add anything other than the integer answer after "La respuesta es ".\n\n{question}"

fr: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "La réponse est ". Do not add anything other than the integer answer after "La réponse est ".\n\n{question}"

ja: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "答えは ". Do not add anything other than the integer answer after "答えは ".\n\n{question}"

th: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "คำตอบคือ ". Do not add anything other than the integer answer after "คำตอบคือ ".\n\n{question}"

zh: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "答案是 ". Do not add anything other than the integer answer after "答案是 ".\n\n{question}"

ar: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "هو باوجلا ". Do not add anything other than the integer answer after "هو باوجلا ".\n\n{question}"

ko: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "답변은 ". Do not add anything other than the integer answer after "답변은 ".\n\n{question}"

pt: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "A resposta é ". Do not add anything other than the integer answer after "A resposta é ".\n\n{question}"

vi: "Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "Câu trả lời là ". Do not add anything other than the integer answer after "Câu trả lời là ".\n\n{question}"

Figure 15: This figure presents the EN prompt for the MGSM dataset.

EN prompt for MLOGIQA:
All: "Passage: {context}\nQuestion: {question}\nChoices:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nPlease choose the most suitable one among A, B, C and D as the answer to this question, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C and D."

Native prompt for MLOGIQA:
zh: "段落: {context}\n问题: {question}\n选择:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n请在 A、B、C 和 D 中选择最合适的一个作为此问题的答案，并以以下 JSON 格式返回：\n{'answer': '[choice]'}\n其中 [choice] 必须是 A、B、C 和 D 中的一项。"

en: "Passage: {context}\nQuestion: {question}\nChoices:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nPlease choose the most suitable one among A, B, C and D as the answer to this question, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C and D."

vi: "Đoạn văn: {context}\nCâu hỏi: {question}\nLựa chọn:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nVui lòng chọn câu trả lời phù hợp nhất trong số A, B, C và D cho câu hỏi này, và trả lại nó trong định dạng JSON sau:\n{'answer': '[choice]'}\ntrong đó [choice] phải là một trong A, B, C và D."

th: "ข้อความ: {context}\nคำถาม: {question}\nตัวเลือก:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nโปรดเลือกขอที่เหมาะสมที่สุดจาก A, B, C และ D เป็นคำตอบของคำถามน และส่งคืนในรูปแบบ JSON ดังต่อไปนี้:\n{'answer': '[choice]'}\nโดยที่ [choice] จะต้องเป็นหนึ่งใน A, B, C และ D."

ar: "الخلفية: {premise}\nالخيارات: \nA. {option_1}\nB. {option_2}\nC. {option_3}\nD. {option_4}\nالمهمة هي اختيار الخيار الأكثر ملاءمة من A و B و C و D، وإرجاع النتيجة بتنسيق JSON التالي:\n{'answer': '[choice]'}\n[choice] يجب أن يكون واحدًا من A أو B أو C أو D."

es: "Pasaje: {context}\nPregunta: {question}\nOpciones:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nPor favor, elija la más adecuada entre A, B, C y D como respuesta a esta pregunta, y devuélvala en el siguiente formato JSON:\n{'answer': '[choice]'}\ndonde [choice] debe ser uno de A, B, C y D."

ja: "本文: {context}\n質問: {question}\n選択肢:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nこの質問の答えとして A、B、C、D の中から最も適したものを選択し、次の JSON 形式で返してください：\n{'answer': '[choice]'}\nここで [choice] は A、B、C、または D のいずれかでなければなりません。"

ko: "구문: {context}\n질문: {question}\n선택:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n이 질문의 답으로 A, B, C 및 D 중 가장 적합한 것을 선택하고, 다음 JSON 형식으로 반환하십시오:\n{'answer': '[choice]'}\n여기서 [choice]는 A, B, C 및 D 중 하나여야 합니다."

fr: "Passage : {context}\nQuestion : {question}\nChoix :\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nVeuillez choisir le plus approprié parmi A, B, C et D comme réponse à cette question, et le renvoyer dans le format JSON suivant :\n{'answer': '[choice]'}\noù [choice] doit être l'un de A, B, C ou D."

pt: "Passagem: {context}\nPergunta: {question}\nOpções:\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\nPor favor, escolha a mais adequada entre A, B, C e D como resposta a esta pergunta, e retorne-a no seguinte formato JSON:\n{'answer': '[choice]'}\ndonde [choice] deve ser uma das opções A, B, C ou D."

Figure 16: This figure presents the prompt for the MLogiQA dataset.

EN prompt for MMMLU:
All: "The following is a multiple-choice question. Please choose the most suitable one among A, B, C and D as the answer to this question, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C and D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

Native prompt for MMMLU:
zh: "以下是一个多项选择题。请在 A、B、C 和 D 中选择最合适的一个作为此问题的答案，并以以下 JSON 格式返回：\n{'answer': '[choice]'}\n其中 [choice] 必须是 A、B、C 和 D 中的一项。\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

en: "The following is a multiple-choice question. Please choose the most suitable one among A, B, C and D as the answer to this question, and return it in the following JSON format:\n{'answer': '[choice]'}\nwhere [choice] must be one of A, B, C and D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

vi: "Dưới đây là một câu hỏi trắc nghiệm. Vui lòng chọn câu trả lời phù hợp nhất trong số A, B, C và D cho câu hỏi này, và trả lại nó trong định dạng JSON sau:\n{'answer': '[choice]'}\ntrong đó [choice] phải là một trong A, B, C và D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

th: "ต่อไปนี้คือคำถามแบบเลือกตอบหลายตัวเลือ โปรดเลือกข้อที่เหมาะสมที่สุดจาก A, B, C และ D เป็นคำตอบของคำถามน และส่งคืนในรูปแบบ JSON ต่อไปนี้:\n{'answer': '[choice]'}\nโดยที่ [choice] จะต้องเป็นหนึ่งใน A, B, C และ D。\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

ar: "التالي هو سؤال متعدد الخيارات. يرجى اختيار الأنسب من بين A و B و C و D كإجابة على هذا السؤال، وإعادته بال تنسيق JSON التالي:\n{'answer': '[choice]'}\nحيث يجب أن يكون [choice] واحدًا من A و B و C و D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

es: "Lo siguiente es una pregunta de opción múltiple. Por favor, elija la más adecuada entre A, B, C y D como respuesta a esta pregunta, y devuélvala en el siguiente formato JSON:\n{'answer': '[choice]'}\ndonde [choice] debe ser uno de A, B, C y D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

ja: "以下は選択式の質問です。この質問の答えとして A、B、C、D の中から最も適したものを選択し、次の JSON 形式で返してください：\n{'answer': '[choice]'}\nここで [choice] は A、B、C、D のいずれかでなければなりません。\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

ko: "다음은 객관식 질문입니다. 이 질문의 답으로 A, B, C 및 D 중 가장 적합한 것을 선택하고 다음 JSON 형식으로 반환하십시오:\n{'answer': '[choice]'}\n여기서 [choice]는 A, B, C 및 D 중 하나여야 합니다.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

fr: "Ce qui suit est une question à choix multiple. Veuillez choisir la plus appropriée parmi A, B, C et D comme réponse à cette question, et la renvoyer dans le format JSON suivant :\n{'answer': '[choice]'}\noù [choice] doit être l'un de A, B, C ou D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

pt: "O seguinte é uma questão de múltipla escolha. Por favor, escolha a mais adequada entre A, B, C e D como resposta a esta pergunta, e retorne-a no seguinte formato JSON:\n{'answer': '[choice]'}\ndonde [choice] deve ser uma das opções A, B, C ou D.\n\n{question}\nA. {option_a}\nB. {option_b}\nC. {option_c}\nD. {option_d}\n"

Figure 17: This figure presents the prompt for the MMMLU dataset.