# OLMES: A Standard for Language Model Evaluations

**Yuling Gu** [α] **Oyvind Tafjord** [α] **Bailey Kuehl** [α] **Dany Haddad** [α]
**Jesse Dodge** [α] **Hannaneh Hajishirzi** [αβ]

[α]Allen Institute for Artificial Intelligence [β]University of Washington
{yulingg, oyvindt}@allenai.org

## Abstract

Progress in AI is often demonstrated by new models claiming improved performance on tasks measuring model capabilities. Evaluating language models can be particularly challenging, as choices of how a model is evaluated on a task can lead to large changes in measured performance. There is no common standard setup, so different models are evaluated on the same tasks in different ways, leading to claims about which models perform best not being reproducible. We propose OLMES, a completely documented, practical, open standard for reproducible LLM evaluations. In developing this standard, we identify and review the varying factors in evaluation practices adopted by the community – such as details of prompt formatting, choice of in-context examples, probability normalizations, and task formulation. In particular, OLMES supports meaningful comparisons between smaller base models that require the unnatural "cloze" formulation of multiple-choice questions against larger models that can utilize the original formulation. OLMES includes well-considered, documented recommendations guided by results from existing literature as well as new experiments resolving open questions.[1]

| Model↓ | Ref1 | Ref2 | Ref3 | Ref4 | Ref5 | Ref6 | OLMES |
|---|---|---|---|---|---|---|---|
| MPT-7B | 47.7 | 42.6 | | | 46.5 | | 45.7 |
| RPJ-Incite-7B | 46.3 | | | | 42.8 | | 45.3 |
| Falcon-7B | 47.9 | 42.4 | | 44.5 | 47.5 | | 49.7 |
| Mistral-7B | 60.0 | | 55.5 | 54.9 | | | 78.6† |
| Llama2-7B | 53.1 | 45.9 | 43.2 | 45.9 | 48.5 | 53.7† | 54.2 |
| Llama2-13B | 59.4 | 49.4 | 48.8 | 49.4 | | 67.6† | 67.3† |
| Llama3-8B | 60.2 | | | | | 78.6† | 79.3† |
| Num shots | 25 | 0 | 0 | 0 | 0 | 25 | 5 |
| Curated shots | No | | | | | No | Yes |
| Formulation | CF | CF | CF? | CF | CF | MCF | MCF/CF |
| Normalization | char | char | ? | char? | pmi | none | none/pmi |

| Ref | Reference citation |
|---|---|
| **Ref1** | HF Open LLM Leaderboard (Beeching et al., 2023) |
| **Ref2** | Llama2 paper (Touvron et al., 2023a) |
| **Ref3** | Mistral 7B (Jiang et al., 2023) |
| **Ref4** | Falcon paper (Almazrouei et al., 2023) |
| **Ref5** | OLMo paper (Groeneveld et al., 2024) |
| **Ref6** | Llama3 model card (AI@Meta, 2024) |

Table 1: Scores reported in different references for LLM performances on ARC-CHALLENGE. Scores indicated with † are using multiple-choice formulation (MCF) rather than "cloze" formulation (CF) (see Section 2.1 for definitions). Entries with "?" denote either undocumented or mixed approaches across models. Different references use different evaluation setups, some of which are not fully specified, so conclusions about performances and relative strengths of models are not reproducible.

## 1 Introduction

Scientific credibility in AI rests on reproducible and well-considered comparisons between models. Many current AI models, such as pretrained large language models (LLMs), are generalist models capable of performing downstream tasks they were not specifically trained on (Brown et al., 2020; Bommasani et al., 2022). When evaluating LLMs on such tasks, there are many choices in how the task is presented to the model and how the model outputs are interpreted before scoring (Gao, 2021; Biderman et al., 2024; Liang et al., 2023). There is currently no standard way to decide on these choices, and they can have significant impact on model performance, with some recent papers claiming as much as an 80% difference in accuracy on a given task just from varying formatting and in-context examples (Sclar et al., 2023).

These choices in evaluation setups are often not reported with enough details to reproduce, so when a team of ML practitioners releases a new model it is often impossible for them to directly compare against previously-reported results by others. Efforts like the Holistic Evaluation of Language Mod-

[1]All prompts, examples, and code used for OLMES are available at https://github.com/allenai/olmes.



*Findings of the Association for Computational Linguistics:*
*NAACL 2025*, pages 5020–5048

els (HELM) benchmark (Liang et al., 2023) and the Hugging Face Open LLM Leaderboard (Beeching et al., 2023) tackle the issue of reproducibility by striving towards standardizing LM evaluations. While the same setup is used to evaluate many models ensuring consistency and reproducibility, the rationale behind prompt formatting, use of in-content examples, normalization techniques, and task formulation are not always clearly documented and thus not consistently followed by other researchers in subsequent work (Touvron et al., 2023b; Biderman et al., 2023; Jiang et al., 2023; Groeneveld et al., 2024; AI@Meta, 2024).

We highlight two problems in the field today. (1) Releasing a new model and comparing it against previously reported results is flawed unless the previous work explicitly described their full evaluation setup, and then that setup is followed in the new work. Currently, different references (like leaderboards, papers) use different (and sometimes underspecified) evaluation setups, leading to different results and conclusions. For a particular dataset, evaluating a specified language model, different references can tell you very different stories. We illustrate this phenomenon in Table 1, which shows how several models' published performance on the ARC-Challenge (Clark et al., 2018) task can vary in the literature. For instance, looking at Ref1, we would conclude that Llama2-13B and Llama3-8B are performing similarly, but Ref6 reveals there is likely a gap of over 10% between them. (2) Despite current efforts to standardize model evaluation (e.g., HF Open LLM Leaderboard, HELM), the choices made are not justified and most model creators do not use these setups for their evaluations. We also see evidence of this in Table 1, showing a variety of setups being used for the same task, differing in choices such as number of shots, source of in-context examples, task formulation, and probability normalization. While these different choices are made in implementing the evaluations, to date, there is no documented standard studying and/or justifying if one choice is better than another, leading to the lack of a set of justified choices that the community can adopt. Mai and Liang (2024) also demonstrates how this might be a community-wide problem in a recent effort (see Figure 4 in Appendix).

To address these problems, we present OLMES (Open Language Model Evaluation Standard), a standard to improve the transparency and reproducibility of language model evaluation from a practical point of view, removing ambiguity in how a final performance metric is obtained when evaluating a model on a dataset. OLMES can be applied to evaluation during the model development process, and in published leaderboards and papers. OLMES provides justified recommendations on all aspects of task setups, such as data sampling, how to format instances, the choice of in-context examples, probability normalization, and task formulation.

Importantly, OLMES is:

- **Reproducible:** OLMES specifies all details of the evaluations, from processing datasets to presenting the task to model, to processing models' outputs, so there are no ambiguities in the evaluation procedure.
- **Practical:** OLMES makes practical decisions in use of computation resources for easy adoption by the community.
- **Documented:** Each decision in the standard is documented with justifications by applying principles from existing studies and performing experiments to resolve open questions.
- **Open**: We release all prompts and code, along with the rationales behind the choices made in OLMES, for subsequent work to follow and build upon by extending the same principles to any new task and model.

Since OLMES is a documented, practical, open evaluation standard, it is straightforward to adopt in publicly available, well-maintained evaluation code bases like the Eleuther LM Evaluation Harness (Gao et al., 2023; Biderman et al., 2024) and HELM (Liang et al., 2023). When used by model developers and other researchers, OLMES will help unify evaluation practices in the field. We believe this work is the first of its kind to unify practices for evaluating base models throughout the full development cycle, from small to large models as well as early to late training stages. All prompts, examples, and code used for OLMES can be found at `https://github.com/allenai/olmes`.

## 2 Experimental setup

### 2.1 Multiple-choice QA and LLM evaluation

Multiple-choice question answering (MCQA) tasks present a compelling way of evaluating models and humans alike, due to the ease of scoring (whether

the correct answer is chosen out of the given options) and the allowed flexibility in the domain and complexity of the questions. One motivation for multiple-choice tasks is that early in training, and for smaller base models before instruction-tuning, other tasks (generative tasks, math reasoning, coding, etc) tend to provide less useful signals. Multiple-choice tasks are the most common type of benchmarks for evaluating base LLMs (Beeching et al., 2023; Touvron et al., 2023a; Jiang et al., 2023; Groeneveld et al., 2024; AI@Meta, 2024), where the evaluation seems straightforward (did the model predict the right answer?), but in practice, a statement like "model X scores Y on ARC-CHALLENGE" is generally uninterpretable (with unspecified details and cannot be meaningfully compared across references, see Table 1) without a clear evaluation standard like OLMES.

We specifically focus on evaluation using these tasks to provide useful guidance during and after base model training, giving important insights into the potential of such models before committing to further tuning (e.g., instruction-tuning). Such tasks form a large, essential part of LLM evaluations and are the focus of OLMES. There are generally two ways to formulate these tasks.

**MCF (Multiple-choice formulation):** presenting answer choices indicated by labels and scoring prediction of answer labels, just like how MCQA is posed to humans. Here is an example of MCQA from ARC-EASY (Clark et al., 2018), a dataset of real grade-school level science questions:

```
Question: Earth's core is primarily composed of which of the following materials?
 A. basalt
 B. iron
 C. magma
 D. quartz
Answer: B
```

**CF (Completion/cloze formulation):** scoring each answer choice separately using LLM token probabilities. The MCF format is not natural for the pure language modeling task of generating the next token. Therefore, the CF format was introduced when evaluating the GPT-3 model (Brown et al., 2020). They found that it was possible to elicit much better performance using a "cloze" completion version of the task, where the model is shown a prompt like:

```
Question: Earth's core is primarily composed of which of the following materials?
Answer: <answer>
```

Each answer choice is separately substituted in for `<answer>`. Then the LLM probability of the answer choice tokens are used to rank the choices and predict an answer. This formulation has ambiguities in how to normalize the probability, as well as absolute limitations, such as not being able to properly address cases where one answer choice is "none of the above" or similar.

### 2.2 Targeted tasks

We select and implement standards for 10 popular benchmark MCQA tasks, see Table 2 for the list. The list covers tasks that are frequently used in the community's evaluation practices, such as the Hugging Face Open LLM Leaderboard (Beeching et al., 2023), Llama papers (Touvron et al., 2023a,b; AI@Meta, 2024), HELM (Liang et al., 2023), and the OLMo evaluation suite (Groeneveld et al., 2024). This selection includes questions on science, various types of commonsense, factual knowledge, and covers a range of topics (MMLU alone covers 57 subjects), of varying difficulty.

### 2.3 Selection of models

We develop OLMES based on a selection of 15 diverse, openly available pretrained LLMs, focusing on base (not instruction-tuned) models, covering a range of sizes from 1B to 70B – Pythia-1B, Pythia-6.7B (Biderman et al., 2023), OLMo-1B, OLMo-7B, OLMo-7B-0424 (Groeneveld et al., 2024), TinyLlama-1.1B (Zhang et al., 2024), StableLM2-1.6B (Bellagente et al., 2024), RPJ-INCITE-7B (Together Computer, 2023), MPT-7b (MosaicML, 2023), Falcon-7B (Almazrouei et al., 2023), Llama2-7B, Llama2-13B (Touvron et al., 2023b), Mistral-7B-v0.1 (Jiang et al., 2023), Llama3-8B, Llama3-70B (AI@Meta, 2024). This reflects our goal of providing an evaluation standard that suits a range of model capabilities, with the flexibility to apply the same methodology during model development as well as when comparing final powerful base models.

Assessing base models of different strengths is important during the training of models and before it is used for further tuning (e.g., instruction-tuning). This is critical for the community when picking between alternate base models for further training or tuning for their application. There is limited established protocol in the community – evaluation during training is often left underspecified and understudied, and when evaluating final

| task | split | #C | # inst (total) | CF norm | reference |
|---|---|---|---|---|---|
| ARC-CHALLENGE (ARC_C) | Test | 4† | 1172 | pmi | (Clark et al., 2018) |
| ARC-EASY (ARC_E) | Test | 4† | 1000 (2376) | char | (Clark et al., 2018) |
| BOOLQ | Val | 2 | 1000 (3270) | none | (Clark et al., 2019) |
| COMMONSENSEQA (CSQA) | Val | 5 | 1221 | pmi | (Talmor et al., 2019) |
| HELLASWAG (HSwag) | Val | 4 | 1000 (10042) | char | (Zellers et al., 2019) |
| MMLU | Test | 4 | 14042 | char | (Hendrycks et al., 2021) |
| OPENBOOKQA (OBQA) | Test | 4 | 500 | pmi | (Mihaylov et al., 2018) |
| PIQA | Val | 2 | 1000 (1838) | char | (Bisk et al., 2020) |
| SOCIAL IQA (SIQA) | Val | 3 | 1000 (1954) | char | (Sap et al., 2019) |
| WINOGRANDE (WinoG) | Val | 2 | 1267 | none | (Sakaguchi et al., 2020) |

Table 2: **OLMES** details on tasks, with our standardized choices of dataset split, number of instances to use (along with total number if sampling was used), and which CF normalization scheme to use (see Section 3.3). Column **#C** shows the number of answer choices (ARC-CHALLENGE and ARC-EASY† have a few instances with 3 or 5 answer choices). See Section 3 for details on instance formatting, choice of in-context examples and task formulation.

base models, researchers across the field use different evaluation setups, leading to different results and conclusions (Tables 1 and 14). We hope this work will empower the community towards more unified practices in benchmarking base models so that further progress can be made on a stronger foundation based on careful evaluation.

## 3 Standardizing variations in evaluation

To evaluate a model on a dataset, there are a variety of decisions that have to be made to get a final score of that model on that dataset. These include:

- How to format dataset instances? (Section 3.1)
- Which few-shot examples to use? (Section 3.2)
- How to normalize LLM probabilities for CF? (Section 3.3)
- What task formulation to use, MCF or CF? (Section 3.4)
- Other implementation choices impacting results (Section 3.5)

Below we enumerate key variations in these steps, and justify the choices made in OLMES (some of which are summarized in Table 2) to standardize these steps, leaving some of the details for the Appendix.

### 3.1 How to format dataset instances?

Each MCQA dataset includes a set of fields used to specify an instance, such as question, answer choices, and perhaps a context for the question. When formatting an instance as a prompt to an LLM, many different choices have been made in the literature. This includes simple choices like `"Question:"` vs `"Q:"` as question prefix (varying even within a paper, e.g., Brown et al. (2020)), or formatting the answer labels (e.g., `"A."` (Touvron et al., 2023a), `"(A)"` (Nori et al., 2023), `"<mc>A</mc>"` (Anthropic, 2024), etc). There is also a choice of whether or not to provide a general instruction, e.g., common for MMLU (Hendrycks et al., 2021), sometimes done for OPENBOOKQA (Almazrouei et al., 2023).

**Instance formatting. OLMES** uses a consistent `"Question: <question>"` prefix and `"Answer:"` suffix in formatting the datasets. This clarifies the question-answering task in a natural way, without relying on verbose instruction understanding. The three exceptions are listed and explained here. For PIQA, we use `"Goal: <goal>"` as the prefix instead to be consistent with the original semantics of the dataset. In the case of MCF, for HELLASWAG, we skip the question prefix and instead add `"Choose the best continuation:"` before presenting the continuation options, and for WINOGRANDE we use the prefix `"Fill in the blank:"` to align with the task. For HELLASWAG and WINOGRANDE, where the CF answer string is simply a language continuation, we remove such prefixes and suffixes for the CF evaluation so that the task is closer to pure language modeling.

**MCQA label choice.** For MCF answer choices, OLMES uses the canonical letters A/B/C/... as answer labels, presenting the multiple-choice options after simple letter labels, i.e., `" A."` format. We note that most tokenizers treat a letter at the start of a line (or string) as a separate token from the same letter following a space. Therefore we add a prefix space in front of each answer label `"\n A. <choice>"` (rather than

"\nA. <choice>"), to work naturally with all current tokenizers (so that the final answer token will be identical to the answer choice token, see Appendix C.3 for details). All the exact OLMES prompt formats are listed in Appendix H.

**Sampling.** Following existing LLM evaluation standardization efforts (Liang et al., 2023; Beeching et al., 2023), **OLMES** uses the test split of a dataset if the labels are publicly available, otherwise the validation split. If the dataset has more than 1500 instances, we sample 1000 instances to evaluate,[2] similar to HELM (Liang et al., 2023) which caps evaluation instances at 1000.[3] Note that the potential extra statistical signal from more instances would generally be dominated by other sources of score variations, like prompt formatting, so this is a practical consideration to avoid unnecessary computation resources. See Table 2 for details on splits and sampling used in OLMES.

### 3.2 Which few-shot examples to use?

Popularized by Brown et al. (2020), it is customary to provide examples of the task to the model through few-shot examples, as this is an effective and universal way to convey a task to an LLM. For example, the MMLU task (Hendrycks et al., 2021) originally came with a fixed 5-shot prompt which is generally used in evaluation (Beeching et al., 2023; Gemma Team et al., 2024; Jiang et al., 2023; Touvron et al., 2023b; AI@Meta, 2024) resulting in more reproducible results than many other tasks.[4] For other tasks, both the number of shots and the way in which they are sampled have varied in different evaluation setups. For example, to evaluate on HELLASWAG, Beeching et al. (2023) sampled 10-shot whereas HELM (Liang et al., 2023) uses 0-shot; within Beeching et al. (2023), a range of 25-shot, 10-shot, 5-shot was sampled for ARC-CHALLENGE, HELLASWAG and WINOGRANDE respectively.

**OLMES** standardizes a manually curated 5-shots prompt for each task (from its training set), ensuring that the examples are of good quality and cover the label space in a balanced way (e.g., avoiding 4 A's and 1 B among the 5 answers).[5] Restricting to 5 in-context examples helps limit computational overhead, similar to HELM (Liang et al., 2023). Analysis suggests that going beyond 5 shots generally does not provide meaningful differences in scores (Brown et al., 2020; Barton, 2024). The manually curated shots for each task can be downloaded from https://github.com/allenai/olmes.

### 3.3 How to normalize LLM probabilities for CF?

When using the completion/cloze formulation (CF) for multiple-choice questions, the LLM returns $P(a_i|q)$, the probability for an answer choice $a_i$ given a question prompt $q$. Ranking solely based on the probability may heavily favor shorter answers with fewer tokens. To work around this issue, different normalization methods have been used in the literature, which we categorize below:

- **none:** $\ln(P(a_i|q))$
- **token:** $\ln(P(a_i|q))/\text{num\_tokens}(a_i)$, which normalizes the log-probability by the number of tokens in the answer (Brown et al., 2020).
- **character:** $\ln(P(a_i|q))/\text{num\_characters}(a_i)$, which normalizes the log-probability by the number of characters in the answer, used by Llama models (Touvron et al., 2023a) and Eleuther AI LM Harness (Gao et al., 2023; Biderman et al., 2024).
- **pmi:** $\ln(P(a_i|q)/P(a_i|u))$ where $u$ ="Answer:" is an unconditional prompt, which normalizes by dividing by the LLM probability of the same answer string without the presence of the question. This can be considered a form of pointwise-mutual-information (PMI) and was explored further in other works (Holtzman et al., 2021).

Efforts like Liang et al. (2023); Gao et al. (2023); Biderman et al. (2024) compare and support comparisons of different normalization approaches, leaving it an open question as to how to make a decision. See Appendix C.2 for further discussions around different normalizations.

To choose a normalization scheme in **OLMES**, we evaluate the models on each dataset, comparing

[2] Sampling uses a specific random seed in Python: Random(1234).sample(all_instances, 1000)

[3] https://crfm-helm.readthedocs.io/en/latest/reproducing_leaderboards/

[4] Sometimes sampled examples are used also for MMLU (MosaicML, 2024). Even for MMLU, noticeable discrepancies have been found, due to other differences in prompt formatting (Mai and Liang, 2024).

[5] More details on curating the examples can be found in Appendix G.

| task | win percentage none | char | tok | pmi | diff oracle | **OLMES** |
|---|---|---|---|---|---|---|
| ARC_C | 0.0 | 33.3 | 0.0 | 66.7 | 0.2 | pmi |
| ARC_E | 6.7 | 86.7 | 6.7 | 0.0 | 0.1 | char |
| BoolQ | 46.7 | 46.7 | 0.0 | 6.7 | 1.1 | none |
| CSQA | 6.7 | 33.3 | 6.7 | 53.3 | 0.6 | pmi |
| HSwag | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | char |
| MMLU | 0.0 | 46.7 | 0.0 | 53.3 | 0.4 | char |
| OBQA | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | pmi |
| PIQA | 6.7 | 46.7 | 46.7 | 0.0 | 0.2 | char |
| SIQA | 0.0 | 86.7 | 6.7 | 6.7 | 0.1 | char |
| WinoG | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | none |

Table 3: Summary of CF normalization comparisons. "win percentage" shows how often each normalization was best across the 15 models. "diff oracle" (difference between the OLMES recommendation and the empirically best normalization for each task and model) shows that there is in general minimal difference between the OLMES normalization and the oracle optimal normalization for each task (difference out of 100%).

the effect of the 4 normalization techniques. Table 3 shows for each task, how often each normalization is empirically the best across the 15 models. Detailed scores per model are in Appendix C.2.

OLMES specifies the **"pmi"** normalization for ARC-CHALLENGE, COMMONSENSEQA, and OPENBOOKQA. The answer choices in these datasets tend to contain unexpected words or phrases that are less likely for models to generate (e.g., "Whirlpool bath" compared to "Bathtub"). The pmi normalization adjusts for this by taking into account the a priori likelihood of the answers. This is consistent with other findings (Holtzman et al., 2021) and some existing evaluation practices, e.g., Brown et al. (2020) selectively uses this normalization for ARC and OPENBOOKQA, and Touvron et al. (2023a,b) for OPENBOOKQA. Computing the extra unconditional likelihood incurs some computation overhead, thus OLMES avoids this normalization for other datasets where there is no strong empirical or theoretical reason to choose this approach.

OLMES specifies the **"character"** normalization for ARC-EASY, HELLASWAG, PIQA, SOCIAL IQA and MMLU. Based on our experiments, it is empirically the normalization technique that gives the best scores[6] for these datasets, and less computationally expensive than the "pmi" normalization. It also has the advantage (unlike the "token" normalization) of already being implemented (as `acc_norm`) in the Eleuther LM Evaluation Harness, where it is generally available for multiple-choice tasks (Gao et al., 2023; Gao, 2021). It is also used in the Hugging Face Open LLM Leaderboard (Beeching et al., 2023) for ARC-CHALLENGE and HELLASWAG, in Touvron et al. (2023a,b)'s evaluations as the default, (with select datasets as exceptions), as well as reported in various works like Biderman et al. (2023); Almazrouei et al. (2023).

OLMES specifies the **"none"** normalization for BOOLQ and WINOGRANDE. In BOOLQ the only answer choices are "yes" or "no" which are single tokens, therefore no length normalization is needed. Note that for some models, the "character" normalization has slightly better performance on BOOLQ (see Table 10), an accidental side effect of "yes" having one more character than "no". One could argue that the pmi normalization is appropriate as it counters any existing bias in the model for "yes" vs "no", but we argue that models should be capable of producing such common words (also indicated in the 5-shot examples) without any such corrections. Finally, WINOGRANDE is a special case in that the continuations are identical (and the prompts vary), so the choice of normalization does not matter and we simply use the **"none"** normalization.

In general, we observe little difference between the OLMES recommendation and the empirically best ("oracle") normalization for each task and model, see "diff oracle" column in Table 3 (Table 9 in Appendix C.2 has more details).

### 3.4 What task formulation to use, MCF or CF?

As LLMs have gotten stronger, the MCQA task formats have gradually changed from CF to MCF. For instance, ARC-CHALLENGE was often evaluated using the CF approach (Touvron et al., 2023a,b; Almazrouei et al., 2023; Beeching et al., 2023), but has switched to MCF for stronger models like OpenAI (2024); AI@Meta (2024), appearing with identical names like "25-shot ARC-CHALLENGE". As an example, AI@Meta (2024) reports a 25-shot MCF ARC-CHALLENGE score for the Llama-3 8B model of 78.6% vs 60.2% for the 25-shot CF on the Hugging Face Open LLM Leaderboard. As performance on a multiple-choice task gets closer to 100%, the CF approach lags behind due to its inherent limitations, giving significantly less signal about a model's actual performance. On the other hand, MMLU is almost exclusively evaluated using the MCF approach, which often results in near-random performance for weaker models (Beeching et al., 2023).

[6]Tie for PIQA, and second-best for MMLU.

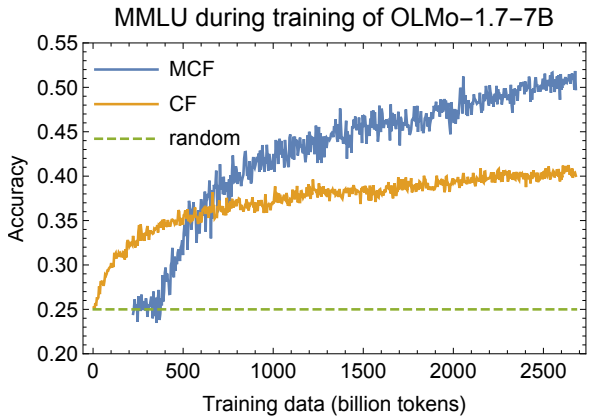


Figure 1: Performance on MMLU validation set during the training of OLMo-7B-0424 model. During early training, there is good signal from CF while MCF is random. Around 400B tokens, the model starts gaining the ability on the MCF format, becoming a stronger signal than CF.

In **OLMES**, we argue that the CF formulation provides a useful evaluation of task knowledge for models that have not yet acquired the skill of answering multiple-choice questions using MCF. On the other hand, MCF is a more realistic formulation for models that can "understand" this format, yielding higher and more representative scores (Robinson et al., 2023; OpenAI, 2024). See Appendix C.1 for further discussion.

We can see an explicit example of a model acquiring the "understanding" of MCF during training in Figure 1, showing the OLMo-7B-0424 model (Groeneveld et al., 2024; AI2 blog, 2024) evaluated on the MMLU validation set in both CF and MCF variations. The plot suggests that model starts learning the MCF task format after about 400 billion training tokens, so in early training CF provides a better signal, while MCF is significantly better in late training where CF levels off.

To further study this phenomenon, we evaluate the CF and MCF versions for each task and model.[7] Figure 2 shows for each task, the MCF and CF performances for the 15 models ordered along the x-axis by overall performance on all tasks. For instance, on ARC-CHALLENGE, we see a clear distinction where the weakest 8 models have near-random performance on the MCF version of the task, yet above random when using CF which offers a better signal to the relative strength of models. For the stronger models, the MCF version clearly outscores the CF version, and is a much better representation of task performance compared to the flatter trends using CF (for Llama3-70B the MCF score is 93.7% (6.3% error) while the CF score is just 69.0% (31% error), a nearly 5x difference in error rate!).

A similar pattern can be seen across other tasks in Figure 2, where the stronger models show performance using MCF either exceeding CF (like ARC-EASY, OPENBOOKQA, MMLU, SOCIAL IQA, COMMONSENSEQA, and PIQA) or at least catching up to it (HELLASWAG, WINOGRANDE, BOOLQ).[8]

In **OLMES**, we standardize to evaluate each model using both the MCF and CF formulations, and the best performing one is used. This allows for meaningful comparison of task evaluation numbers over a range of models, from the smaller, weaker base models which can only deal with the CF (where MCF scores hovering around random baseline), to the stronger models which can report more accurate performance using the MCF (where CF provides less clear signal).

### 3.5 Other implementation details

There are other important details that go into a fully specified evaluation result, and we enumerate the choices made in OLMES here:

- For MMLU: use macro average (over 57 tasks) rather than micro average (over 14042 instances), following AI@Meta (2024). This better represents the diversity of fields in the dataset, although in practice it does not generally make a big difference (see Figure 8).
- When a model requires it, make sure to add the appropriate <bos> token at start of prompt (e.g., Gemma (Gemma Team et al., 2024)).
- When using the "character" normalization for CF, include the leading space in the calculation of answer length.
- Restrict all inputs (with completions) to 2048 tokens for consistency across models.[9]
- Use the default model precision when evaluating (i.e., avoid options like `load_in_8bit` unless it produces identical results).
- OLMES uses the standard approach of two newlines to separate each in-context example.
- Other than the original instruction line for MMLU (Hendrycks et al., 2021), we do not

[7] More detailed numbers can be found in Tables 6 and 7 in the Appendix B.

[8] Appendix C.2.1 provides further discussion.

[9] For current tasks this is only exhausted for a few MMLU instances.

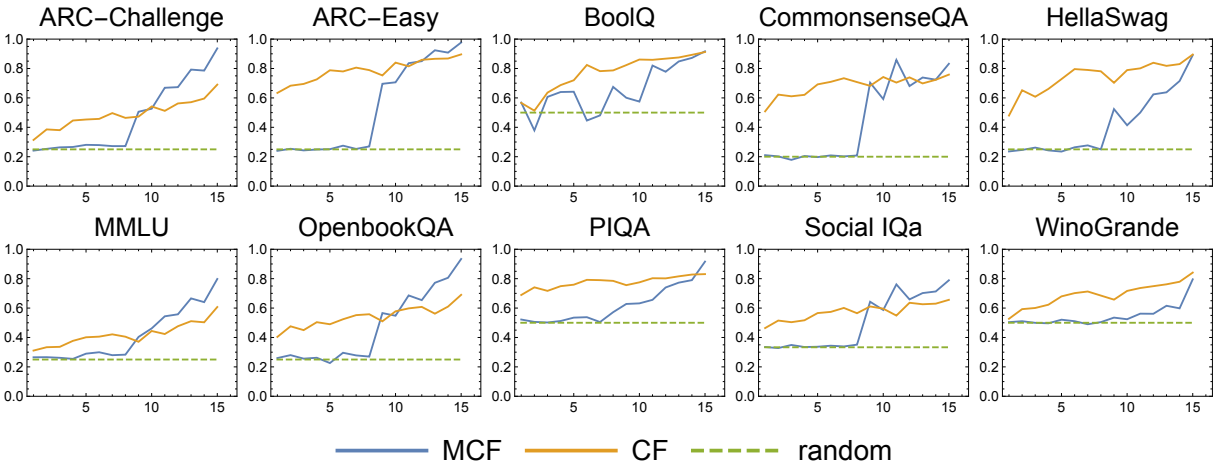


Figure 2: Comparing model performance on each task for MCF vs CF. The 15 models are ordered along the x-axis by overall performance across all 10 tasks. In general, CF is needed to elicit a non-random signal from the weaker models, while stronger models can take advantage of MCF for a more accurate assessment.

add any extra instructions. This is in view of previous work finding the subject information from instructions makes little changes to model ranking (Alzahrani et al., 2024), and to reduce additional sources of variation in the prompt.

Note that computational details, like batch size and type/state of GPU, can affect floating point operations such that answer choice decisions can flip if they are very close. This is hard to avoid unless one considers "ties" when answers are sufficiently close in confidence, we leave that for future consideration. A reference implementation of OLMES is released at `https://github.com/allenai/olmes` under the Apache 2.0 license.

## 4 OLMES: Summary and results

OLMES includes the following elements, justified in detail above:

- Use test set when available, otherwise validation. Sample 1000 instances if more than 1500 (Section 3.1)
- Use specified, exact prompt format (Section 3.1)
- Use fixed, curated 5-shot examples (Section 3.2)
- Use prescribed probability normalization for CF (Section 3.3)
- Evaluate with both MCF and CF, use the best result (Section 3.4)
- Follow recommendations for all other evaluation details (in Section 3.5)

Table 4 reports the overall, fully reproducible, OLMES scores for the 15 models across the benchmarks. An extended table with a total of 40 models is shown in Table 13 (Appendix D).

## 5 Related work

With model releases, performance on popular datasets is used to gauge the progress achieved e.g., OpenAI (2024) showing superhuman performance on benchmarks like MMLU. Such evaluation also guides community efforts towards understanding and sharing findings on what it takes to build a strong model (Touvron et al. (2023a,b); Biderman et al. (2023); Almazrouei et al. (2023); MosaicML (2023); Jiang et al. (2023); Gemma Team et al. (2024); Groeneveld et al. (2024) *inter alia*). However, given a model and a dataset, even for the frequently used datasets, there are varied practices in how accuracy on them is measured. Various work has shown that model evaluations are vulnerable to differences such as option position changes in multiple-choice questions (Zheng et al., 2024; Li et al., 2024), choice symbols, re-ordering of answer options, changing number of answer options (Wang et al., 2024), and task formulation (Alzahrani et al., 2024; Robinson et al., 2023; Khatun and Brown, 2024; Wiegreffe et al., 2023). Even minor formatting changes can cause large, generally arbitrary, score variations (Sclar et al., 2023).

The Holistic Evaluation of Language Models (HELM) benchmark (Liang et al., 2023), the Hug-

| model | ARC_C | ARC_E | BoolQ | CSQA | HSwag | MMLU | OBQA | PIQA | SIQA | WinoG | average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Pythia-1B | 31.4 | 63.4 | 56.8† | 50.9 | 48.0 | 31.1 | 40.4 | 68.9 | 46.4 | 52.7 | 49.0 |
| OLMo-1B | 38.6 | 68.3 | 51.3 | 62.2 | 65.2 | 33.4 | 47.6 | 74.1 | 51.5 | 59.3 | 55.1 |
| TinyLlama-1.1B | 38.1 | 69.5 | 63.6 | 61.1 | 60.8 | 33.6 | 45.0 | 71.7 | 50.4 | 60.1 | 55.4 |
| Pythia-6.7B | 44.6 | 72.6 | 68.7 | 62.1 | 66.1 | 37.7 | 50.4 | 74.9 | 51.7 | 62.3 | 59.1 |
| RPJ-INCITE-7B | 45.3 | 78.8 | 72.0 | 69.2 | 72.8 | 40.1 | 49.0 | 75.9 | 56.6 | 68.0 | 62.8 |
| StableLM2-1.6B | 50.6† | 75.3 | 82.3 | 70.4† | 70.3 | 40.4† | 56.6† | 75.6 | 64.3† | 65.7 | 65.1 |
| OLMo-7B | 46.4 | 78.9 | 78.7 | 70.8 | 78.1 | 40.5 | 55.8 | 78.5 | 56.5 | 68.5 | 65.3 |
| MPT-7b | 45.7 | 78.0 | 82.4 | 70.9 | 79.6 | 40.6 | 52.4 | 79.2 | 57.4 | 70.2 | 65.6 |
| Falcon-7B | 49.7 | 80.6 | 78.2 | 73.4 | 79.0 | 42.1 | 55.2 | 79.0 | 60.1 | 71.3 | 66.9 |
| Llama2-7B | 54.2 | 84.0 | 86.1 | 74.2 | 78.9 | 46.2† | 57.8 | 77.5 | 59.6 | 71.7 | 69.0 |
| Llama2-13B | 67.3† | 85.9 | 86.7 | 74.0 | 83.9 | 55.8† | 65.4† | 80.2 | 65.9† | 74.9 | 74.0 |
| OLMo-7B-0424 | 66.9† | 83.6† | 85.9 | 85.8† | 80.1 | 54.4† | 68.6† | 80.3 | 76.1† | 73.6 | 75.5 |
| Llama3-8B | 79.3† | 92.4† | 87.5 | 73.9† | 81.8 | 66.6† | 77.2† | 81.6 | 70.2† | 76.2 | 78.7 |
| Mistral-7B-v0.1 | 78.6† | 90.8† | 89.3 | 72.4† | 83.0 | 64.0† | 80.6† | 82.8 | 71.3† | 77.9 | 79.1 |
| Llama3-70B | 93.7† | 97.7† | 91.7† | 83.2† | 89.5 | 79.8† | 93.4† | 91.6† | 78.9† | 84.1 | 88.4 |

Table 4: Reproducible performance scores across models and tasks using OLMES, providing robust, meaningful comparisons across a wide range of models and tasks. † indicates use of the MCF score.

ging Face Open LLM Leaderboard (Beeching et al., 2023), Mosaic Eval Gauntlet (Barton, 2024), Eleuther LM Evaluation Harness (Gao et al., 2023; Biderman et al., 2024), and Unitxt (Bandel et al., 2024) present efforts toward greater transparency and reproducibility of LLM evaluations. These frameworks generally describe and provide support for various task setups, presenting them as open choices to researchers and users. When specific default setups are given, the rationale is not always documented and thus not followed by others in subsequent work (see Tables 1 and 14).

## 6 Discussion

By identifying and reviewing common evaluation practices in the community, and performing experiments to resolve open questions, we present OLMES – an open, documented, reproducible, and practical evaluation standard. OLMES provides justified recommendations on decisions such as how to format dataset instances, the choice of in-context examples, task formulation, probability normalization, as well as other implementation details. The goal is for OLMES to be a useful guide for model developers to obtain signals as to whether their model is on track during training, and to compare final powerful base models. The practical choices encourage evaluations without unnecessary computation resources. The reproducible nature means that any evaluation done using OLMES can be directly compared to existing OLMES evaluations. We also document the rationales behind the choices made, guiding the community toward more justified evaluation practices. OLMES can be applied to current leaderboards and evaluation code bases to unify evaluation practices in the field.

**Future work and limitations.** Future work includes adding more tasks to OLMES, covering tasks beyond MCQA such as generative tasks and chain-of-thought prompting. This will include standardizing how answers are extracted for evaluation, and for chat models how to split the prompt into messages. We welcome the community to contribute to OLMES, extending the principles of OLMES to new tasks.

OLMES is a step towards standardizing LLM evaluations, ready to be incorporated into evaluation code bases for broad usage. OLMES facilitates robust and simplified comparisons of model performances, both for researchers during model training and development, and for developers in choosing models to build upon.

## Acknowledgments

In creating this evaluation standard, OLMES, we build on top of the various previous efforts on language model evaluation in the community – including previous work on language model evaluation standardization, the many open research reports disclosing how evaluation on LLMs have been done, and the datasets that made OLMES possible, which we explicitly cite and acknowledge in our paper.

## Limitations

The current version of OLMES is focused on providing guidance useful for LLM evaluation during the training stage and for comparing final base models, which provides important insights into the potential of such models before further tuning (e.g.,

instruction-tuning, safety-tuning). Interesting directions for future work include looking into evaluations targeted at accessing the effectiveness of various kinds of model tuning, as well as evaluation for multi-modal models.

This paper focuses on design choices in evaluating language models with multiple-choice tasks. While the suite of multiple-choice tasks used in this work includes questions on science, various types of commonsense, factual knowledge, and covers a range of topics (MMLU alone covers 57 subjects), of varying difficulty, an important future direction would be to apply the same principles in OLMES (e.g., prompt formatting, curated few-shot examples) to generative tasks and chain-of-thought prompting.

While the recommendations in OLMES are well-considered, justified and practical, they do not cover all plausible variants of presenting a task. See Appendix A for further discussion, showing how performance measured using OLMES is stable and consistent when subject to small changes in prompt wording or the selection of few-shot examples, within the general recommendations. Larger differences would be expected when diverging from OLMES recommendations such as by using unnatural prompts e.g., using rare symbols as answer labels, or randomly sampled few-shot examples which could run into skewed label distribution covered in few-shot examples or include noisy examples from train sets. We leave evaluating the robustness of models under adversarial setups as a topic for future work.

## Ethical considerations

This study involves the use of large-scale language models. We only use their outputs to obtain their answers to questions in commonly used multiple-choice datasets, therefore we do not foresee any ethical issues with their use for the research presented in this work.

## References


AI2 blog. 2024. OLMo 1.7–7B: A 24 point improvement on MMLU. https://blog.allenai.org/olmo-1-7-7b-92b43f7d269d. Accessed: 2024-06-03.

AI@Meta. 2024. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. Accessed: 2024-05-29.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *arXiv:2311.16867*.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *arXiv:2402.01781*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf. Accessed: 2024-06-03.

Elron Bandel, Yotam Perlitz, Elad Venezian, Roni Friedman, Ofir Arviv, Matan Orbach, Shachar Don-Yehiya, Dafna Sheinwald, Ariel Gera, Leshem Choshen, Michal Shmueli-Scheuer, and Yoav Katz. 2024. Unitxt: Flexible, shareable and reusable data preparation and evaluation for generative AI. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 207–215, Mexico City, Mexico. Association for Computational Linguistics.

Tessa Barton. 2024. Calibrating the mosaic evaluation gauntlet. https://www.databricks.com/blog/calibrating-mosaic-evaluation-gauntlet. Accessed: 2024-05-05.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, Meng Lee, Emad Mostaque, Michael Pieler, Nikhil Pinnaparju, Paulo Rocha, Harry Saini, Hannah Teufel, Niccolo Zanichelli, and Carlos Riquelme. 2024. Stable LM 2 1.6b technical report. *arXiv:2402.17834*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri

Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv:2405.14782*.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. *arXiv:2108.07258*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

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. 2022. PaLM: Scaling language modeling with pathways. *arXiv:2204.02311*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, arXiv:1803.05457.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Leo Gao. 2021. Multiple choice normalization in LM evaluation. `https://blog.eleuther.ai/multiple-choice-normalization/`. Accessed: 2024-05-08.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023.

A framework for few-shot language model evaluation. https://zenodo.org/records/10256836.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *arXiv:2403.08295*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. *arXiv:2402.00838*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv:2310.06825*.

Aisha Khatun and Daniel G. Brown. 2024. A study on large language models' limitations in multiple-choice question answering. *arXiv:2401.07955*.

Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. 2024. Can multiple-choice questions really be useful in detecting the abilities of LLMs? In *LREC-COLING 2024*.

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 Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad 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 S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew 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*.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*.

Yifan Mai and Percy Liang. 2024. Massive multitask language understanding (MMLU) on helm. https://crfm.stanford.edu/2024/05/01/helm-mmlu.html. Accessed: 2024-05-29.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

MosaicML. 2023. Introducing MPT-7B: A new standard for open-source, commercially usable LLMs. https://www.databricks.com/blog/mpt-7b. Accessed: 2024-05-08.

MosaicML. 2024. Mosaic eval gauntlet v0.3.0 - evaluation suite. https://github.com/mosaicml/llm-foundry/blob/main/scripts/eval/local_data/EVAL_GAUNTLET.md. Accessed: 2024-05-29.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of GPT-4 on medical challenge problems. *arXiv:2303.13375*.

OpenAI. 2024. GPT-4 technical report. *arXiv:2303.08774*.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv:2201.11990*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Together Computer. 2023. RedPajama: an open dataset for training large language models.

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. 2023a. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*.

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

Haochun Wang, Sendong Zhao, Zewen Qiang, Bing Qin, and Ting Liu. 2024. Beyond the answers: Reviewing the rationality of multiple choice question answering for the evaluation of large language models. *arXiv:2402.01349*.

Sarah Wiegreffe, Matthew Finlayson, Oyvind Tafjord, Peter Clark, and Ashish Sabharwal. 2023. Increasing probability mass on answer choices does not always improve accuracy. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8392–8417, Singapore. 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 Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. TinyLlama: An open-source small language model. *arXiv:2401.02385*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. *Proceedings of the International Conference on Learning Representations (ICLR)*.

## A FAQs

**Q: Are there existing established protocols that differ from the proposed ones? Will they cause resistance from the research community in adopting the OLMES standard?**

The lack of "established protocols" is precisely the issue that our work addresses. As discussed in our Introduction and Related work sections, existing efforts either present **different variations as open choices to users** or are **underspecified** or **under-documented (e.g., the rationale behind default choice is unexplained)** and thus not followed by others in subsequent work. Therefore, there is no existing established protocol in the community – researchers across the field use different evaluation setups, with reasons behind their choices left unexplained, leading to different results and conclusions. We illustrate this in Table 1, with Table 14 showing an extended version to include score variations across different references on OpenbookQA in addition to ARC-Challenge.

This is the main motivation for the standard, to reconcile the differences in practices so that scores reported in papers can be meaningfully interpreted and compared (as we show, a statement like "score on 25-shot ARC Challenge" is woefully underspecified, whereas "score on ARC Challenge using OLMES" is a well-defined number without any ambiguity). We also justify each decision we make so that the community can, for the first time, appreciate the rationale behind the setups and thus encourage broad adoption.

**Q: What is novel about OLMES?**

Building on the many existing works that introduce new methodologies (e.g., new way of prompting, probability normalization, etc), OLMES is the first work of its kind to provide a **completely open, practical, reproducible, and documented evaluation standard with justified choices** so that results across research work can be meaningfully compared. This fills an important gap in current research on LLMs – the adoption of OLMES by model developers and other researchers will help unify evaluation practices in the field for the first time, significantly shifting current research paradigms.

**Q: Why is OLMES more principled than trying a range of settings?**

Rather than having to train a model from scratch to discover patterns in task formulation, run the different settings to choose a normalization scheme, or delve into the same literature again to study the variants, the community can now directly build upon the various choices in OLMES.

We hope to guide the community towards more well-documented and justifiable chosen evaluation settings like OLMES without having to go through trying a mix of less informed choices (which we argue should be avoided altogether). Through extensive literature review and experimentation, we observe that some settings provide better signals than others, and document them in this work to guide the community to use them, a few examples include:

- CF gives a clearer signal early in training, which is helpful for developers to cheaply make modeling decisions. On the other hand, MCF is a better indicator of performance later on. CF often works better for weaker models while MCF is at random, and MCF is a better representation of task performance for stronger models.
- Few-shot prompting is an effective and universal way to convey a task to an LLM (more stable learning curve than 0-shot) but going beyond 5 shots generally does not provide meaningful differences in scores.
- For probability normalization, in BoolQ the only answer choices are "yes" or "no" which are single tokens, therefore no length normalization is needed. Even if for some models, the "character" normalization has slightly better performance on BoolQ (see Table 9), one should note that this is an accidental side effect of "yes" having one more character than "no" and indeed a normalization which changes the probability of "yes" vs "no" simply because the "no" token has fewer characters seem problematic.

Not only are recommendations in OLMES backed by both existing literature and new experimental results, there is also little difference between the OLMES recommendation and the empirically best ("oracle") normalization for each task and model, see "diff oracle" column. We argue that adopting such an approach is a better practice than blindly optimizing for the best performance e.g., problematically using "character" normalization for BOOLQ.

**Q: Including a broader range of datasets?**

The focus on multiple-choice datasets in OLMES is motivated by their frequent use in evaluating base LLMs, where the evaluation seems straightforward (did the model predict the right answer?) but in practice, a statement like "model X scores Y on ARC Challenge" is generally uninterpretable (with unspecified details and cannot be meaningfully compared across references) without a clear evaluation standard like OLMES. In this paper, we focused on datasets chosen to provide guidance useful for the training stage and evaluation of base models, which provides important insights into the potential of such models before further tuning (e.g., instruction-tuning).

Note that the fundamental principles of OLMES as introduced, generalize to any dataset of interest. Rather than viewing what we have illustrated in our paper as a fixed set, our goal is to use that as an illustration and empower researchers to move towards reproducible evaluation by applying OLMES to any dataset of interest suited for their own work.

**Q: Evaluating on more models? What are some valuable insights from extended experiments?**

We provide additional results in Appendix D, Table 13 with additional models. Evaluating different models using OLMES provides valuable insights for understanding LLMs and model development. For instance, within each batch of model release by developers, models of bigger size perform better than smaller ones (see average scores of Pythia-6.7B outperforms Pythia-1B, OLMo-7B outperforms OLMo-1B, Llama2-13B outperforms Llama2-7B, Gemma2-9B outperforms Gemma2-2B). However, size is not the only way to get to a stronger model, evaluating on OLMES also allow the community meaningful comparison of models to witness the effect of model improvement via better training data, model architecture, as well as other improved approaches as researchers iterate on their models e.g., OLMo-7B-0424's improvement over initial OLMo-7B; Llama3-8B's improvement over Llama2-7B and even Llama2-13B, Mixtral-8x7B-v0.1 outperforming the Mistral model, aligning with the insights reported in these model releases documenting their improved recipes and innovations for better models. Further, OLMES also gives meaningful comparison of model performance as researchers experiment to reduce computational costs, e.g., our results align with the original DeepSeekMoE paper where they "scale up DeepSeekMoE to 16B parameters and show that it achieves comparable performance with LLaMA2 7B, with only about 40% of computations". All these underscore the applicability and value of OLMES in supporting unified evaluation as the field progress towards better models, as an open, well-documented, practical and reproducible evaluation standard. We make all prompts, examples, and code used for OLMES openly available, and encourage researchers to try it for any model of their interest be it one they are studying or building.

**Q: How does OLMES stay relevant in the rapid evolution of AI and LLMs?**

We have been continuously looking out for new LLMs and evaluating them using OLMES, showing that the same guiding principles still apply as best practices providing a systematic, comparable approach. See extended evaluation results in Table 13.

As the field moves forward, we look forward to applying the principles of OLMES to more benchmarks and evaluating newer models using OLMES. While we are working on extending OLMES, we do not anticipate revisions to the currently established recommendations in OLMES any time soon as the guiding principles are built on top of a rich literature of existing work over the years and will likely remain relevant in the community for a while in the near future.

**Q: Why not use prompting techniques such as CoT or self-reflection?**

While these prompting strategies have shown to be useful for instruction-tuned models, they tend to be much less effective on base models, which is a focus of this work. E.g., some experiments we performed with MMLU showed that various CoT prompts (both zero-shot and few-shot) have a positive boost on instruction-tuned models (like Llama-3.1-8B-Instruct), but tend to lower the scores a bit for base models (like Llama-3.1-8B).

**Q: How do you ensure that formatting settings are fair to all models?**

Supported by reviewing common evaluation practices in the community and empirical evidence across a wide range of models, the recommendations we make are at least as reasonable and fair as the myriad of settings that have been used in the literature.

If a model is peculiar in any specific way (e.g., only able to do multiple choice questions with one type of answer label like "1." or "2."), it is not the goal of OLMES to tailor to such peculiarities as this standard is intended to be applied across a range of models and to encourage the development of models that produce reasonable outputs given any reasonable input.

**Q: What happens when there are minor variants to OLMES?**

Through OLMES, we provide best practices to evaluate language models and justify our choices. Our choices are mostly aligned with common practices in LLM evaluations, but with defining standards in formatting, choice of in-context examples, probability normalizations, and task formulation. In the process, we accounted for many factors, taking into consideration the robustness of OLMES under minor variations. We discuss some of these considerations here.

**[Part 1] Order of presenting the options A/B/C/D:**

The order of presenting the multiple-choice options A/B/C/D does not apply to CF since each answer is processed independently. For MCF it is indeed a confounder that some (especially weaker) models might highly prefer a given label (like B). The benchmarks in OLMES are generally balanced such that such a model would not be much better than random. Further, if this happens, CF would generally get a better score in such cases and OLMES would use that score in its final output. Therefore, having a setting where we use both CF (not affected by the order of options) and MCF (where the order of options may matter) makes sure the final metric will not be hugely affected by such factors. We considered applying more rigorous measures (like running all cyclic permutations of answer choices) but decided for practical reasons, the extra processing time and complexity were not worth the minor improvements in robustness (as one consideration of OLMES is also to be a practical standard that does not take unnecessarily more compute than is needed).

**[Part 2] Minor variations in prompt wording or few-shot examples:**

To address potential concerns on minor variations in prompt wording or few-shot examples, we evaluated under three additional settings, while adhering to the general principles in OLMES:

**Variant 1 (minor variation in prompting):**
3 changes to OLMES prompt format - (1) change the label and text separator from "." to ")", (2) insert an additional new line before the answer descriptor, (3) change the "Answer" descriptor to "Correct answer"
**Variant 2 (varying few-shot examples):**
Create a different set of curated few-shot examples by changing 3 out of the 5 in-context examples to new ones that are different from those in OLMES. In picking the new few-shot examples, the same recommendations were followed to ensure diversity in the examples and that they cover the label space.
**Variant 3 (minor variation in prompting + varying few-shot examples):**
Apply changes in both Variants 1 and 2 together.

| model | ARC_E orig | var 1 | var 2 | var 3 | avg | diff | std err |
|---|---|---|---|---|---|---|---|
| Pythia-1B | 63.4 | 63.3 | 62.7 | 62.7 | 63.0 | 0.4 | 1.5 |
| Llama2-7B | 84.0 | 84.4 | 83.4 | 84.4 | 84.0 | 0.0 | 1.2 |
| DeepSeek-7B | 80.6 | 80.9 | 80.5 | 80.4 | 80.6 | 0.0 | 1.3 |
| Gemma2-2B | 84.3† | 83.2† | 83.9† | 82.8† | 83.5 | 0.8 | 1.2 |
| Llama3-8B | 92.4† | 92.5† | 92.3† | 93.1† | 92.6 | 0.2 | 0.8 |
| **model** | **OBQA orig** | **var 1** | **var 2** | **var 3** | **avg** | **diff** | **std err** |
| Pythia-1B | 40.4 | 38.6 | 39.4 | 37.6 | 39.0 | 1.4 | 2.2 |
| Llama2-7B | 57.8 | 57.2 | 55.2 | 57.4 | 56.9 | 0.9 | 2.2 |
| DeepSeek-7B | 62.2† | 61.0† | 61.6† | 63.2† | 62.0 | 0.2 | 2.2 |
| Gemma2-2B | 68.8† | 67.2† | 68.8† | 67.0† | 68.0 | 0.8 | 2.1 |
| Llama3-8B | 77.2† | 76.8† | 78.8† | 77.8† | 77.7 | 0.5 | 1.9 |
| **model** | **PIQA orig** | **var 1** | **var 2** | **var 3** | **avg** | **diff** | **std err** |
| Pythia-1B | 68.9 | 69.2 | 69.2 | 69.3 | 69.1 | 0.2 | 1.5 |
| Llama2-7B | 77.5 | 77.2 | 77.7 | 77.8 | 77.5 | 0.0 | 1.3 |
| DeepSeek-7B | 79.3 | 78.8 | 80.9 | 80.6 | 79.9 | 0.6 | 1.3 |
| Gemma2-2B | 78.5 | 77.8 | 79.3 | 78.5 | 78.5 | 0.0 | 1.3 |
| Llama3-8B | 81.6 | 80.7 | 82.4 | 82.8 | 81.9 | 0.3 | 1.2 |

Table 5: Extended results comparing using OLMES (orig) and when the setting is subjected to minor variations in prompt wording (var1), few-shot examples (var2), or both (var3). † indicates the use of MCF. The "avg" score obtained via averaging orig, var1, var2, and var3 results is often within 1% of that obtained by the original OLMES setup (orig). We report the observed differences between averaging the 4 setups ("avg") and directly using OLMES (orig) in the "diff" column, illustrating the minor differences (often <1%) do not justify the 4 times more compute needed, against the "practical" consideration in OLMES.

We report these additional results in Table 5. Following EleutherAI in calculating standard error (https://github.com/EleutherAI/lm-evaluation-harness/blob/ebe7226ebfb8d11a9fb8d6b53eb65891f895c633/lm_eval/api/metrics.py#L288), in the additional results, we also incorporated bounds on standard error in our evaluations using OLMES (see "std err" column). This provides a statistical bound on the degree of variation in reported numbers and illustrates that while any performance metric should be interpreted to have slight variants (e.g., < 2.5%), the scenario where a model underperforms significantly due to minor variants is unlikely statistically.

The additional results show that differences in performance between averaging variations vs. using the OLMES setup directly were generally minimal, typically less than 1 percent (the largest difference seen is 1.4%). This suggests that performance measured using OLMES is quite stable and consistent when subject to small changes in prompt wording or the selection of few-shot examples, within the general recommendations. Note that these variants still format the instances in natural ways and are slight modifications of the original settings of OLMES, still adhering to the general principles such as instance formatting that clarifies the task in a natural way and choice of in-context examples to cover a range of examples and different answer labels. Larger differences would be expected when diverging from OLMES recommendations such as by using unnatural prompts e.g., using rare symbols as answer labels, or randomly sampled few-shot examples which could run into skewed label distribution covered in few-shot examples or include noisy examples from train sets. We observe that current successful language models are generally robust to the OLMES evaluation standard. OLMES has been informed by prior efforts like HELM and Eleuther LM Evaluation Harness, therefore the prompts are designed to be natural, and suitable for evaluating language models.

## B Detailed CF and MCF task scores

Tables 6 and 7 present detailed scores across all tasks, with both MCF and CF results (using the OLMES recommendations for CF normalization).

## C Further details on variations

In this appendix we discuss further details on how LLM evaluations can vary and the choices made in OLMES.

### C.1 Task formulation details

LLM evaluations started out using the CF approach for many tasks (Brown et al., 2020; Du et al., 2022; Smith et al., 2022; Chowdhery et al., 2022; Lieber et al., 2021), which is a more reasonable option for weaker models that struggle with the more natural MCF (Khatun and Brown, 2024). The task formulation only very recently and gradually switched to the MCF approach when it became clear that the model could utilize it, producing higher scores (Robinson et al., 2023; OpenAI, 2024; AI@Meta, 2024).

The HELM study (Liang et al., 2023) included comparisons between the MCF ("joint") and CF ("separate") approaches, finding that certain models can really benefit from the MCF approach, although among the models in the original study it was really only the Anthropic-LM v4-s3 (52B) model which could take full advantage of it.

### C.2 CF normalization details

Tables 10, 11 and 12 show detailed comparisons of CF normalization on different models, for the various tasks.

Unlike in MCF, where the evaluation metric involves just scoring the log-likelihood corresponding to the answer choice label (i.e., A/B/C/...), there is a choice of log-likelihood normalization ("none", "per token", "per character" or "pmi") for CF as detailed in Section 3.3.

When evaluating the GPT-3 model (Brown et al., 2020), they worked around this issue by normalizing the log-probability by the number of tokens in the answer (similar to how loss is computed during training). They also noted that for a few datasets, it worked markedly better to instead "normalize" by dividing by LLM probability of the same answer string without the presence of the question (usually by just having a generic prefix like `"Answer: <answer_string>"`). This can be considered a form of pointwise-mutual-information (PMI) and was explored further in other works (Holtzman et al., 2021).

The Eleuther LM Evaluation Harness (Gao et al., 2023; Biderman et al., 2024) and some subsequent evaluations (e.g., the Llama models (Touvron et al., 2023a)) have also used "per answer character" normalization, using the argumentation (Gao, 2021; Biderman et al., 2024), that normalizing per token is problematic since it depends on the tokenizer. Since the purpose of the normalization is simply to rank the answer choices within themselves (keeping model and tokenizer fixed), this does not seem like a relevant argument, and indeed a normalization which changes the probability of "yes" vs "no" simply because the "no" token has fewer characters seem problematic. In practice, for tasks where answers are either relatively long or similar in length, there are minor differences between these two length normalizations.

The HELM study (Liang et al., 2023) included comparisons between these normalization approaches for a number of tasks and models (using the terms "separate" and "separate calibrated" for "token" and "pmi" respectively), eventually settling on a default choice for each, not unlike the choices in the GPT-3 report (Brown et al., 2020). The Eleuther LM Evaluation Harness generally reports two metrics for each multiple-choice task: `acc` (using the "none" normalization) and `acc_norm` (using the "character" normalization).

#### C.2.1 Tasks that generally prefer CF

HellaSwag and WinoGrande continue to have CF scores higher than MCF scores even for the strongest models that can understand the MCF prompt. This somewhat surprising tendency seems correlated with the fact that these tasks in the CF format are exactly like the language modeling task of finding the most natural continuation of a running piece of text. Judging from the trends in the plot, it would also be

| model | ARC_C | | ARC_E | | BoolQ | | CSQA | | HSwag | | MMLU | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MCF | CF | MCF | CF | MCF | CF | MCF | CF | MCF | CF | MCF | CF |
| Pythia-1B | 24.1 | 31.4 | 24.0 | 63.4 | 56.8 | 56.6 | 21.0 | 50.9 | 23.6 | 48.0 | 26.5 | 31.1 |
| OLMo-1B | 25.3 | 38.6 | 25.4 | 68.3 | 37.9 | 51.3 | 20.2 | 62.2 | 24.6 | 65.2 | 26.6 | 33.4 |
| TinyLlama-1.1B | 26.4 | 38.1 | 24.3 | 69.5 | 60.7 | 63.6 | 17.9 | 61.1 | 26.2 | 60.8 | 26.2 | 33.6 |
| Pythia-6.7B | 26.6 | 44.6 | 24.9 | 72.6 | 64.0 | 68.7 | 20.5 | 62.1 | 24.3 | 66.1 | 25.4 | 37.7 |
| RPJ-INCITE-7B | 28.1 | 45.3 | 25.1 | 78.8 | 64.2 | 72.0 | 19.7 | 69.2 | 23.5 | 72.8 | 29.0 | 40.1 |
| StableLM2-1.6B | 50.6 | 47.3 | 69.6 | 75.3 | 60.1 | 82.3 | 70.4 | 68.2 | 52.4 | 70.3 | 40.4 | 37.1 |
| OLMo-7B | 27.2 | 46.4 | 27.0 | 78.9 | 67.5 | 78.7 | 20.8 | 70.8 | 25.0 | 78.1 | 28.3 | 40.5 |
| MPT-7b | 27.9 | 45.7 | 27.5 | 78.0 | 44.6 | 82.4 | 20.9 | 70.9 | 26.4 | 79.6 | 30.0 | 40.6 |
| Falcon-7B | 27.2 | 49.7 | 25.3 | 80.6 | 48.1 | 78.2 | 20.2 | 73.4 | 27.7 | 79.0 | 28.0 | 42.1 |
| Llama2-7B | 52.6 | 54.2 | 70.6 | 84.0 | 57.5 | 86.1 | 59.2 | 74.2 | 41.4 | 78.9 | 46.2 | 44.4 |
| Llama2-13B | 67.3 | 56.2 | 85.0 | 85.9 | 77.8 | 86.7 | 68.1 | 74.0 | 62.4 | 83.9 | 55.8 | 47.6 |
| OLMo-7B-0424 | 66.9 | 51.2 | 83.6 | 81.5 | 82.0 | 85.9 | 85.8 | 70.4 | 50.0 | 80.1 | 54.4 | 42.4 |
| Llama3-8B | 79.3 | 57.1 | 92.4 | 86.6 | 84.8 | 87.5 | 73.9 | 69.9 | 63.8 | 81.8 | 66.6 | 51.1 |
| Mistral-7B-v0.1 | 78.6 | 59.6 | 90.8 | 86.8 | 87.2 | 89.3 | 72.4 | 72.3 | 71.5 | 83.0 | 64.0 | 50.3 |
| Llama3-70B | 93.7 | 69.0 | 97.7 | 89.6 | 91.7 | 91.2 | 83.2 | 75.8 | 89.1 | 89.5 | 79.8 | 60.7 |

Table 6: Comparing MCF and CF scores on each task (part 1). Weaker models at the top of the table have near-random MCF scores, while for stronger models at the bottom, the MCF score provides a better assessment than the CF score.

| model | OBQA | | PIQA | | SIQA | | WinoG | | average scores | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MCF | CF | MCF | CF | MCF | CF | MCF | CF | MCF | CF | all | max |
| Pythia-1B | 26.0 | 40.4 | 52.2 | 68.9 | 33.5 | 46.4 | 50.4 | 52.7 | 33.8 | 49.0 | 41.4 | 49.0 |
| OLMo-1B | 28.0 | 47.6 | 50.6 | 74.1 | 32.8 | 51.5 | 51.1 | 59.3 | 32.3 | 55.1 | 43.7 | 55.1 |
| TinyLlama-1.1B | 25.6 | 45.0 | 50.2 | 71.7 | 34.9 | 50.4 | 50.0 | 60.1 | 34.2 | 55.4 | 44.8 | 55.4 |
| Pythia-6.7B | 26.2 | 50.4 | 51.2 | 74.9 | 33.5 | 51.7 | 49.6 | 62.3 | 34.6 | 59.1 | 46.9 | 59.1 |
| RPJ-INCITE-7B | 22.6 | 49.0 | 53.5 | 75.9 | 33.7 | 56.6 | 52.1 | 68.0 | 35.1 | 62.8 | 49.0 | 62.8 |
| StableLM2-1.6B | 56.6 | 51.0 | 62.8 | 75.6 | 64.3 | 61.1 | 53.5 | 65.7 | 58.1 | 63.4 | 60.7 | 65.1 |
| OLMo-7B | 27.0 | 55.8 | 57.2 | 78.5 | 35.1 | 56.5 | 50.4 | 68.5 | 36.6 | 65.3 | 50.9 | 65.3 |
| MPT-7b | 29.6 | 52.4 | 53.8 | 79.2 | 34.4 | 57.4 | 51.1 | 70.2 | 34.6 | 65.6 | 50.1 | 65.6 |
| Falcon-7B | 27.8 | 55.2 | 50.5 | 79.0 | 33.9 | 60.1 | 49.0 | 71.3 | 33.8 | 66.9 | 50.3 | 66.9 |
| Llama2-7B | 54.8 | 57.8 | 63.2 | 77.5 | 58.7 | 59.6 | 52.4 | 71.7 | 55.7 | 68.8 | 62.2 | 69.0 |
| Llama2-13B | 65.4 | 60.8 | 74.0 | 80.2 | 65.9 | 63.6 | 56.1 | 74.9 | 67.8 | 71.4 | 69.6 | 74.0 |
| OLMo-7B-0424 | 68.6 | 59.8 | 65.6 | 80.3 | 76.1 | 54.9 | 56.2 | 73.6 | 68.9 | 68.0 | 68.5 | 75.5 |
| Llama3-8B | 77.2 | 56.2 | 77.3 | 81.6 | 70.2 | 62.6 | 61.6 | 76.2 | 74.7 | 71.0 | 72.9 | 78.7 |
| Mistral-7B-v0.1 | 80.6 | 61.0 | 79.0 | 82.8 | 71.3 | 63.0 | 59.8 | 77.9 | 75.5 | 72.6 | 74.1 | 79.1 |
| Llama3-70B | 93.4 | 69.0 | 91.6 | 83.1 | 78.9 | 65.6 | 79.6 | 84.1 | 87.9 | 77.8 | 82.8 | 88.4 |

Table 7: Comparing MCF and CF scores on each task (part 2), along with overall averages. The "max" average corresponds to the OLMES score, taking the best of MCF and CF for each task.

| model | MCF-macro | MCF-micro | CF-macro | CF-micro |
|---|---|---|---|---|
| Pythia-6.7B | 25.4 | 25.2 | 37.7 | 37.5 |
| TinyLlama-1.1B | 26.2 | 25.7 | 33.6 | 33.5 |
| Pythia-1B | 26.5 | 26.4 | 31.1 | 31.2 |
| OLMo-1B | 26.6 | 26.3 | 33.4 | 33.6 |
| Falcon-7B | 28.0 | 27.7 | 42.1 | 41.9 |
| OLMo-7B | 28.3 | 28.3 | 40.5 | 40.7 |
| RPJ-INCITE-7B | 29.0 | 28.4 | 40.1 | 40.1 |
| MPT-7b | 30.0 | 29.3 | 40.6 | 40.6 |
| StableLM2-1.6B | 40.4 | 39.6 | 37.1 | 37.0 |
| Llama2-7B | 46.2 | 45.5 | 44.4 | 44.3 |
| OLMo-7B-0424 | 54.4 | 52.8 | 42.4 | 42.4 |
| Llama2-13B | 55.8 | 55.5 | 47.6 | 47.1 |
| Mistral-7B-v0.1 | 64.0 | 63.0 | 50.3 | 49.8 |
| Llama3-8B | 66.6 | 65.4 | 51.1 | 50.8 |
| Llama3-70B | 79.8 | 79.2 | 60.7 | 60.5 |

Table 8: Macro vs micro average scores on MMLU, where macro average is over the 57 tasks and micro average is over the 14042 individual questions. In general there are small differences between the two.

| | **ARC_C** | | **ARC_E** | | **BoolQ** | | **CSQA** | | **HSwag** | | **MMLU** | | **OBQA** | | **PIQA** | | **SIQA** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| model | pmi | diff | char | diff | none | diff | pmi | diff | char | diff | char | diff | pmi | diff | char | diff | char | diff |
| Pythia-1B | 31.4 | 0.0 | 63.4 | 0.0 | 56.6 | 4.5 | 50.9 | 0.0 | 48.0 | 0.0 | 31.1 | 1.2 | 40.4 | 0.0 | 68.9 | 1.4 | 46.4 | 0.0 |
| OLMo-1B | 38.6 | 0.0 | 68.3 | 0.2 | 51.3 | 4.7 | 62.2 | 0.0 | 65.2 | 0.0 | 33.4 | 0.8 | 47.6 | 0.0 | 74.1 | 0.0 | 51.5 | 0.0 |
| TinyLlama-1.1B | 38.1 | 0.0 | 69.5 | 0.0 | 63.6 | 2.2 | 61.1 | 0.0 | 60.8 | 0.0 | 33.6 | 0.9 | 45.0 | 0.0 | 71.7 | 0.6 | 50.4 | 0.0 |
| Pythia-6.7B | 44.6 | 0.0 | 72.6 | 0.0 | 68.7 | 0.0 | 62.1 | 0.2 | 66.1 | 0.0 | 37.7 | 0.2 | 50.4 | 0.0 | 74.9 | 0.0 | 51.7 | 1.1 |
| RPJ-INCITE-7B | 45.3 | 0.0 | 78.8 | 0.0 | 72.0 | 2.5 | 69.2 | 0.2 | 72.8 | 0.0 | 40.1 | 0.8 | 49.0 | 0.0 | 75.9 | 0.1 | 56.6 | 0.0 |
| MPT-7b | 45.7 | 0.6 | 78.0 | 0.0 | 82.4 | 0.0 | 70.9 | 0.0 | 79.6 | 0.0 | 40.6 | 0.0 | 52.4 | 0.0 | 79.2 | 0.0 | 57.4 | 0.0 |
| Falcon-7B | 49.7 | 0.0 | 80.6 | 0.0 | 78.2 | 0.6 | 73.4 | 0.0 | 79.0 | 0.0 | 42.1 | 0.0 | 55.2 | 0.0 | 79.0 | 0.2 | 60.1 | 0.0 |
| OLMo-7B | 46.4 | 0.0 | 78.9 | 0.0 | 78.7 | 0.0 | 70.8 | 0.0 | 78.1 | 0.0 | 40.5 | 0.1 | 55.8 | 0.0 | 78.5 | 0.8 | 56.5 | 0.0 |
| StableLM2-1.6B | 47.3 | 0.0 | 75.3 | 0.0 | 82.3 | 0.0 | 68.2 | 0.0 | 70.3 | 0.0 | 37.1 | 1.5 | 51.0 | 0.0 | 75.6 | 0.3 | 61.1 | 0.0 |
| Llama2-7B | 54.2 | 0.0 | 84.0 | 0.0 | 86.1 | 0.0 | 74.2 | 0.0 | 78.9 | 0.0 | 44.4 | 0.4 | 57.8 | 0.0 | 77.5 | 0.2 | 59.6 | 0.0 |
| OLMo-7B-0424 | 51.2 | 0.0 | 81.5 | 0.0 | 85.9 | 0.0 | 70.4 | 1.1 | 80.1 | 0.0 | 42.4 | 0.0 | 59.8 | 0.0 | 80.3 | 0.0 | 54.9 | 0.8 |
| Llama2-13B | 56.2 | 0.9 | 85.9 | 0.0 | 86.7 | 1.5 | 74.0 | 0.0 | 83.9 | 0.0 | 47.6 | 0.0 | 60.8 | 0.0 | 80.2 | 0.0 | 63.6 | 0.0 |
| Llama3-8B | 57.1 | 1.3 | 86.6 | 0.0 | 87.5 | 0.3 | 69.9 | 4.3 | 81.8 | 0.0 | 51.1 | 0.0 | 56.2 | 0.0 | 81.6 | 0.0 | 62.6 | 0.0 |
| Mistral-7B-v0.1 | 59.6 | 0.6 | 86.8 | 0.0 | 89.3 | 0.0 | 72.3 | 2.1 | 83.0 | 0.0 | 50.3 | 0.0 | 61.0 | 0.0 | 82.8 | 0.0 | 63.0 | 0.0 |
| Llama3-70B | 69.0 | 0.0 | 89.6 | 0.8 | 91.2 | 0.5 | 75.8 | 1.3 | 89.5 | 0.0 | 60.7 | 0.0 | 69.0 | 0.0 | 83.1 | 0.1 | 65.6 | 0.0 |

Table 9: Normalization details, showing that our recommendations are not only supported by reasoning using principles behind the normalization but also close to the empirically best normalization that lets you get the highest accuracy for each model on each task (see "diff" columns).

| | **ARC_C** | | | | | **ARC_E** | | | | | **BoolQ** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| model | none | char | tok | pmi | best | none | char | tok | pmi | best | none | char | tok | pmi | best |
| Pythia-1B | 26.1 | 28.4 | 29.0 | 31.4 | pmi | 61.9 | 63.4 | 60.9 | 56.5 | char | 56.6 | 61.1 | 56.6 | 41.0 | char |
| OLMo-1B | 32.9 | 34.4 | 34.7 | 38.6 | pmi | 68.5 | 68.3 | 65.8 | 60.2 | none | 51.3 | 56.0 | 51.3 | 42.3 | char |
| TinyLlama-1.1B | 31.5 | 34.1 | 32.2 | 38.1 | pmi | 68.6 | 69.5 | 64.4 | 60.4 | char | 63.6 | 65.8 | 63.6 | 53.6 | char |
| Pythia-6.7B | 36.3 | 39.5 | 39.0 | 44.6 | pmi | 71.4 | 72.6 | 70.0 | 64.1 | char | 68.7 | 66.9 | 68.7 | 47.6 | none |
| RPJ-INCITE-7B | 40.3 | 43.5 | 42.9 | 45.3 | pmi | 76.1 | 78.8 | 75.9 | 70.1 | char | 72.0 | 74.5 | 72.0 | 72.4 | char |
| MPT-7b | 41.7 | 46.3 | 44.7 | 45.7 | char | 76.3 | 78.0 | 76.2 | 68.5 | char | 82.4 | 79.9 | 82.4 | 76.7 | none |
| Falcon-7B | 41.6 | 47.4 | 47.6 | 49.7 | pmi | 77.0 | 80.6 | 78.3 | 69.8 | char | 78.2 | 78.8 | 78.2 | 77.6 | char |
| OLMo-7B | 41.6 | 45.5 | 45.0 | 46.4 | pmi | 76.7 | 78.9 | 77.4 | 69.6 | char | 78.7 | 77.7 | 78.7 | 78.6 | none |
| StableLM2-1.6B | 42.2 | 44.3 | 44.9 | 47.3 | pmi | 73.3 | 75.3 | 74.4 | 70.0 | char | 82.3 | 82.0 | 82.3 | 76.1 | none |
| Llama2-7B | 48.4 | 52.0 | 50.2 | 54.2 | pmi | 81.4 | 84.0 | 81.0 | 74.7 | char | 86.1 | 85.6 | 86.1 | 80.5 | none |
| OLMo-7B-0424 | 45.5 | 49.3 | 48.5 | 51.2 | pmi | 79.2 | 81.5 | 79.7 | 71.1 | char | 85.9 | 83.8 | 85.9 | 85.6 | none |
| Llama2-13B | 52.4 | 57.1 | 54.2 | 56.2 | char | 83.9 | 85.9 | 82.8 | 77.6 | char | 86.7 | 88.2 | 86.7 | 77.5 | char |
| Llama3-8B | 53.6 | 58.4 | 56.8 | 57.1 | char | 85.8 | 86.6 | 85.8 | 76.6 | char | 87.5 | 87.8 | 87.5 | 67.0 | char |
| Mistral-7B-v0.1 | 56.1 | 60.2 | 58.9 | 59.6 | char | 84.7 | 86.8 | 84.6 | 78.6 | char | 89.3 | 89.1 | 89.3 | 89.2 | none |
| Llama3-70B | 65.7 | 69.0 | 67.7 | 69.0 | char | 89.7 | 89.6 | 90.4 | 82.6 | tok | 91.2 | 90.4 | 91.2 | 91.7 | pmi |
| **average scores** | 43.7 | 47.3 | 46.4 | 49.0 | NA | 77.0 | 78.7 | 76.5 | 70.0 | NA | 77.4 | 77.8 | 77.4 | 70.5 | NA |
| **win percentage** | 0.0 | 33.3 | 0.0 | 66.7 | **pmi** | 6.7 | 86.7 | 6.7 | 0.0 | **char** | 46.7 | 46.7 | 0.0 | 6.7 | **none** |

Table 10: Comparing CF normalization schemes (part 1).

| model | CSQA | | | | | HSwag | | | | | MMLU | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | none | char | tok | pmi | best | none | char | tok | pmi | best | none | char | tok | pmi | best |
| Pythia-1B | 47.7 | 50.9 | 47.3 | 50.9 | char | 39.2 | 48.0 | 47.8 | 41.0 | char | 29.5 | 31.1 | 30.8 | 32.3 | pmi |
| OLMo-1B | 56.8 | 60.0 | 57.6 | 62.2 | pmi | 50.9 | 65.2 | 64.1 | 49.8 | char | 31.7 | 33.4 | 33.3 | 34.2 | pmi |
| TinyLlama-1.1B | 58.9 | 60.5 | 55.9 | 61.1 | pmi | 46.9 | 60.8 | 59.7 | 48.5 | char | 31.2 | 33.6 | 33.0 | 34.5 | pmi |
| Pythia-6.7B | 59.5 | 62.2 | 58.9 | 62.1 | char | 50.4 | 66.1 | 65.9 | 53.5 | char | 34.9 | 37.7 | 37.0 | 37.9 | pmi |
| RPJ-INCITE-7B | 67.7 | 69.4 | 67.2 | 69.2 | char | 55.7 | 72.8 | 71.8 | 60.6 | char | 37.4 | 40.1 | 40.0 | 40.9 | pmi |
| MPT-7b | 69.6 | 70.3 | 69.1 | 70.9 | pmi | 60.5 | 79.6 | 76.5 | 61.5 | char | 37.8 | 40.6 | 40.1 | 40.4 | char |
| Falcon-7B | 70.0 | 70.3 | 69.5 | 73.4 | pmi | 60.7 | 79.0 | 78.4 | 60.0 | char | 39.3 | 42.1 | 41.9 | 42.1 | char |
| OLMo-7B | 69.0 | 70.0 | 67.9 | 70.8 | pmi | 59.3 | 78.1 | 76.3 | 64.2 | char | 37.9 | 40.5 | 40.5 | 40.6 | pmi |
| StableLM2-1.6B | 63.6 | 66.3 | 65.6 | 68.2 | pmi | 54.7 | 70.3 | 69.7 | 56.4 | char | 35.2 | 37.1 | 37.1 | 38.6 | pmi |
| Llama2-7B | 70.5 | 72.7 | 68.4 | 74.2 | pmi | 61.9 | 78.9 | 77.1 | 64.4 | char | 42.0 | 44.4 | 43.9 | 44.8 | pmi |
| OLMo-7B-0424 | 71.6 | 63.5 | 59.0 | 70.4 | none | 61.4 | 80.1 | 77.7 | 65.2 | char | 39.9 | 42.4 | 42.2 | 41.8 | char |
| Llama2-13B | 72.2 | 72.7 | 68.4 | 74.0 | pmi | 63.7 | 83.9 | 81.0 | 70.3 | char | 44.3 | 47.6 | 46.7 | 47.1 | char |
| Llama3-8B | 72.0 | 74.2 | 73.5 | 69.9 | char | 62.8 | 81.8 | 80.3 | 71.1 | char | 47.5 | 51.1 | 50.8 | 49.6 | char |
| Mistral-7B-v0.1 | 73.1 | 73.8 | 74.4 | 72.3 | tok | 64.5 | 83.0 | 81.0 | 70.3 | char | 46.9 | 50.3 | 50.0 | 49.0 | char |
| Llama3-70B | 77.1 | 77.1 | 77.1 | 75.8 | char | 70.3 | 89.5 | 87.1 | 80.8 | char | 57.2 | 60.7 | 60.5 | 59.4 | char |
| **average scores** | 66.6 | 67.6 | 65.3 | 68.4 | NA | 57.5 | 74.5 | 73.0 | 61.2 | NA | 39.5 | 42.2 | 41.9 | 42.2 | NA |
| **win percentage** | 6.7 | 33.3 | 6.7 | 53.3 | **pmi** | 0.0 | 100.0 | 0.0 | 0.0 | **char** | 0.0 | 46.7 | 0.0 | 53.3 | **pmi** |

Table 11: Comparing CF normalization schemes (part 2)

| model | OBQA | | | | | PIQA | | | | | SIQA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | none | char | tok | pmi | best | none | char | tok | pmi | best | none | char | tok | pmi | best |
| Pythia-1B | 20.2 | 28.6 | 30.4 | 40.4 | pmi | 70.3 | 68.9 | 68.8 | 60.1 | none | 42.8 | 46.4 | 46.0 | 44.4 | char |
| OLMo-1B | 26.0 | 33.0 | 38.4 | 47.6 | pmi | 73.2 | 74.1 | 73.2 | 59.9 | char | 45.3 | 51.5 | 49.9 | 47.3 | char |
| TinyLlama-1.1B | 24.4 | 34.8 | 35.8 | 45.0 | pmi | 72.1 | 71.7 | 72.3 | 62.0 | tok | 45.6 | 50.4 | 48.2 | 48.4 | char |
| Pythia-6.7B | 25.8 | 37.0 | 37.4 | 50.4 | pmi | 74.8 | 74.9 | 74.3 | 63.6 | char | 48.0 | 51.7 | 52.8 | 49.2 | tok |
| RPJ-INCITE-7B | 31.8 | 40.0 | 42.8 | 49.0 | pmi | 74.9 | 75.9 | 76.0 | 61.9 | tok | 50.8 | 56.6 | 56.0 | 52.2 | char |
| MPT-7b | 31.6 | 43.8 | 43.8 | 52.4 | pmi | 77.7 | 79.2 | 78.1 | 63.7 | char | 51.0 | 57.4 | 55.9 | 52.5 | char |
| Falcon-7B | 35.2 | 45.8 | 44.4 | 55.2 | pmi | 78.3 | 79.0 | 79.2 | 63.2 | tok | 52.9 | 60.1 | 57.5 | 54.4 | char |
| OLMo-7B | 33.2 | 42.8 | 45.0 | 55.8 | pmi | 78.2 | 78.5 | 79.3 | 65.2 | tok | 50.3 | 56.5 | 56.5 | 52.8 | char |
| StableLM2-1.6B | 34.4 | 41.6 | 45.2 | 51.0 | pmi | 75.2 | 75.6 | 75.9 | 63.6 | tok | 52.7 | 61.1 | 60.7 | 56.1 | char |
| Llama2-7B | 33.8 | 44.6 | 45.0 | 57.8 | pmi | 76.7 | 77.5 | 77.7 | 62.9 | tok | 52.6 | 59.6 | 58.3 | 53.6 | char |
| OLMo-7B-0424 | 37.2 | 48.4 | 49.6 | 59.8 | pmi | 78.5 | 80.3 | 79.3 | 66.3 | char | 53.5 | 54.9 | 54.3 | 55.7 | pmi |
| Llama2-13B | 39.2 | 46.4 | 48.4 | 60.8 | pmi | 78.9 | 80.2 | 79.8 | 66.4 | char | 56.7 | 63.6 | 60.7 | 56.8 | char |
| Llama3-8B | 37.0 | 47.6 | 50.0 | 56.2 | pmi | 79.7 | 81.6 | 81.1 | 67.5 | char | 54.6 | 62.6 | 60.1 | 56.4 | char |
| Mistral-7B-v0.1 | 38.2 | 48.4 | 50.0 | 61.0 | pmi | 80.8 | 82.8 | 81.3 | 67.4 | char | 55.6 | 63.0 | 60.9 | 57.5 | char |
| Llama3-70B | 47.0 | 55.0 | 56.6 | 69.0 | pmi | 82.8 | 83.1 | 83.2 | 68.3 | tok | 59.7 | 65.6 | 64.8 | 57.3 | char |
| **average scores** | 33.0 | 42.5 | 44.2 | 54.1 | NA | 76.8 | 77.6 | 77.3 | 64.1 | NA | 51.5 | 57.4 | 56.2 | 53.0 | NA |
| **win percentage** | 0.0 | 0.0 | 0.0 | 100.0 | **pmi** | 6.7 | 46.7 | 46.7 | 0.0 | **char** | 0.0 | 86.7 | 6.7 | 6.7 | **char** |

Table 12: Comparing CF normalization schemes (part 3).

interesting to monitor if as even more capable models are developed, the MCF scores will eventually surpass that of the CF scores (given how close they already get to each other).

#### C.2.2 Hybrid formulation

In CF, overall probability score could be quite misleading since it may heavily favor shorter answers with fewer tokens. Note that this would be different if the answer choices are actually listed before scoring the answer string, then most tokens (after the choice has been disambiguated by the first few tokens) would have probability near one. This "hybrid" formulation has been used in some cases, but usually scores in between the CF and MCF approaches (Wiegreffe et al., 2023). However, this hybrid approach is not popular in evaluation standardization efforts like the Open LLM Leaderboard, HELM, or when used to evaluate models during development, so it is not a focus in OLMES.

### C.3 Tokenization of MCQA choice labels

When formatting multiple-choice questions, OLMES specifies the use of a prefix space in front of each answer choice, that is `"\n A. <choice>"` rather than `"\nA. <choice>"`. Figure 3 shows explicit examples of tokenizers where this helps maintain a correspondence between the token for the answer label and the token in the final answer (e.g., `"\nAnswer: A"`). E.g., for the Llama tokenizer, the consistent token is the `"_A"` rather than the separate token `"A"` you get without the prefix space.

```
> from transformers import AutoTokenizer
> llama_tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-2-7b-hf")
> olmo_tokenizer = AutoTokenizer.from_pretrained("allenai/OLMo-7B-0424-hf")
> test_string = "What is 3+4?\n A. 7\nA. 7\nAnswer: A"
> llama_tokenizer.tokenizer(test_string)
['_What', '_is', '_', '3', '+', '4', '?', '<0x0A>', '_A', '.', '_', '7', '<0x0A>', 'A', '.', '_',
'7', '<0x0A>', 'Answer', ':', '_A']
> olmo_tokenizer.tokenizer(test_string)
['What', 'Ġis', 'Ġ3', '+', '4', '?', 'Ċ', 'ĠA', '.', 'Ġ7', 'Ċ', 'A', '.', 'Ġ7', 'Ċ', 'Answer',
':', 'ĠA']
```

Figure 3: Tokenizer example, showing two examples of tokenizers which need a prefix space before MCQA answer choice labels to represent the choice label and the final answer label using the same token.

## D Extended OLMES result table

Table 13 shows OLMES evaluations across an extended set of 40 models. Table 14 shows an extended version of Table 1 which includes score variations across different references on OpenBookQA in addition to ARC-Challenge.

## E HELM Reproduction of MMLU

In Figure 4 we see data taken from HELM's reproduction of MMLU scores for a variety of models.

## F Compute used

The inference on the models evaluated were done on NVIDIA RTX A6000 GPUs. A total of around 400 GPU hours was used.

## G Curation of 5-shot examples: considerations

Procedure for manually curating the few-shot examples:

- Download the train set from Hugging Face datasets
- Start from the beginning of the training set, looking at a batch of 10 (i.e., start with first 10)

| model | ARC_C | ARC_E | BoolQ | CSQA | HSwag | MMLU | OBQA | PIQA | SIQA | WinoG | average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Pythia-1B | 31.4 | 63.4 | 56.8† | 50.9 | 48.0 | 31.1 | 40.4 | 68.9 | 46.4 | 52.7 | 49.0 |
| OLMo-1B-0724 | 36.4 | 53.5 | 66.8 | 42.4 | 67.5 | 32.0 | 44.2 | 74.0 | 45.2 | 62.9 | 52.5 |
| OLMo-1B | 38.6 | 68.3 | 51.3 | 62.2 | 65.2 | 33.4 | 47.6 | 74.1 | 51.5 | 59.3 | 55.1 |
| TinyLlama-1.1B | 38.1 | 69.5 | 63.6 | 61.1 | 60.8 | 33.6 | 45.0 | 71.7 | 50.4 | 60.1 | 55.4 |
| Qwen2-0.5B | 48.4† | 64.9† | 64.3 | 56.2 | 48.9 | 45.3† | 51.6† | 67.9 | 54.7† | 56.1 | 55.8 |
| Llama3.2-1B | 43.5 | 71.6 | 69.4 | 59.6 | 67.3 | 38.2 | 42.0 | 73.7 | 52.0 | 62.5 | 58.0 |
| Pythia-6.7B | 44.6 | 72.6 | 68.7 | 62.1 | 66.1 | 37.7 | 50.4 | 74.9 | 51.7 | 62.3 | 59.1 |
| RPJ-INCITE-7B | 45.3 | 78.8 | 72.0 | 69.2 | 72.8 | 40.1 | 49.0 | 75.9 | 56.6 | 68.0 | 62.8 |
| Gemma-2B | 49.9 | 80.2 | 76.6 | 68.9 | 72.5 | 41.7† | 52.4 | 76.1 | 57.1 | 66.1 | 64.2 |
| StableLM2-1.6B | 50.6† | 75.3 | 82.3 | 70.4† | 70.3 | 40.4† | 56.6† | 75.6 | 64.3† | 65.7 | 65.1 |
| OLMo-7B | 46.4 | 78.9 | 78.7 | 70.8 | 78.1 | 40.5 | 55.8 | 78.5 | 56.5 | 68.5 | 65.3 |
| MPT-7b | 45.7 | 78.0 | 82.4 | 70.9 | 79.6 | 40.6 | 52.4 | 79.2 | 57.4 | 70.2 | 65.6 |
| Zamba2-1.2B | 55.0† | 85.4 | 76.1 | 70.1 | 73.4 | 44.7† | 59.8† | 76.6 | 58.4 | 67.2 | 66.7 |
| Falcon-7B | 49.7 | 80.6 | 78.2 | 73.4 | 79.0 | 42.1 | 55.2 | 79.0 | 60.1 | 71.3 | 66.9 |
| DCLM-1B | 57.6† | 79.5 | 80.9 | 71.3 | 75.1 | 48.5† | 60.0† | 76.6 | 60.5† | 68.1 | 67.8 |
| DeepSeek-MoE-16B | 53.4 | 82.7 | 81.9 | 72.7 | 80.4 | 45.5† | 58.4 | 80.1 | 59.9 | 73.2 | 68.8 |
| Llama2-7B | 54.2 | 84.0 | 86.1 | 74.2 | 78.9 | 46.2† | 57.8 | 77.5 | 59.6 | 71.7 | 69.0 |
| DeepSeek-7B | 57.2† | 80.6 | 84.8 | 74.0 | 80.4 | 48.7† | 62.2† | 79.3 | 65.1† | 72.5 | 70.5 |
| Qwen2-1.5B | 68.6† | 85.2† | 75.3 | 72.0† | 67.6 | 56.5† | 74.6† | 75.7 | 65.3† | 64.5 | 70.5 |
| OLMoE-1B-7B-0924 | 62.1† | 84.2 | 79.2 | 72.9 | 80.0 | 54.1† | 65.4† | 79.8 | 63.0† | 70.2 | 71.1 |
| Gemma2-2B | 67.5† | 84.3† | 83.6 | 66.4† | 74.6 | 53.3† | 68.8† | 78.5 | 64.7† | 71.8 | 71.3 |
| Llama3.2-3B | 69.6† | 85.1† | 78.3 | 69.0 | 77.0 | 57.8† | 67.2† | 77.4 | 64.9† | 69.9 | 71.6 |
| JetMoE-8B | 61.4† | 81.9† | 85.7 | 75.3† | 81.7 | 49.1† | 68.0† | 80.3 | 71.3† | 70.7 | 72.5 |
| Llama2-13B | 67.3† | 85.9 | 86.7 | 74.0 | 83.9 | 55.8† | 65.4† | 80.2 | 65.9† | 74.9 | 74.0 |
| OLMo-7B-0424 | 66.9† | 83.6† | 85.9 | 85.8† | 80.1 | 54.4† | 68.6† | 80.3 | 76.1† | 73.6 | 75.5 |
| OLMo-7B-0724 | 68.0† | 85.7† | 85.3 | 85.4† | 80.5 | 54.9† | 67.6† | 79.3 | 76.1† | 73.2 | 75.6 |
| DeepSeek-V2-Lite | 74.0† | 88.9† | 84.7 | 73.8 | 81.9 | 58.8† | 72.4† | 80.2 | 69.1† | 74.0 | 75.8 |
| Qwen1.5-MoE-A2.7B | 77.4† | 91.6† | 85.0 | 81.4† | 80.0 | 62.4† | 80.6† | 81.0 | 74.1† | 72.3 | 78.6 |
| Llama3-8B | 79.3† | 92.4† | 87.5 | 73.9† | 81.8 | 66.6† | 77.2† | 81.6 | 70.2† | 76.2 | 78.7 |
| Mistral-7B-v0.3 | 78.3† | 91.1† | 88.4 | 72.7† | 83.1 | 63.5† | 80.0† | 81.9 | 71.2† | 77.7 | 78.8 |
| Llama3.1-8B | 79.5† | 91.7† | 88.5 | 74.3† | 81.6 | 66.9† | 78.6† | 81.1 | 71.4† | 76.6 | 79.0 |
| Mistral-7B-v0.1 | 78.6† | 90.8† | 89.3 | 72.4† | 83.0 | 64.0† | 80.6† | 82.8 | 71.3† | 77.9 | 79.1 |
| DCLM-7B | 79.8† | 92.3† | 87.0 | 77.0 | 82.3 | 64.4† | 79.6† | 80.1 | 71.2† | 77.3 | 79.1 |
| Qwen2-7B | 88.1† | 95.3† | 88.9 | 81.2† | 86.4† | 71.8† | 88.2† | 86.0† | 78.0† | 75.1 | 83.9 |
| Gemma2-9B | 89.5† | 95.5† | 89.4 | 78.8† | 87.3† | 70.6† | 88.4† | 86.1† | 76.0† | 78.8 | 84.0 |
| Mixtral-8x7B-v0.1 | 87.1† | 96.1† | 90.0† | 78.3† | 86.7 | 71.9† | 87.0† | 86.1† | 75.1† | 82.6 | 84.1 |
| Zamba2-7B | 92.2† | 96.7† | 89.3 | 84.0† | 89.4† | 68.5† | 84.2† | 86.5† | 77.7† | 79.6 | 84.8 |
| Llama3.1-70B | 92.8† | 97.4† | 91.9 | 81.7† | 89.4 | 79.1† | 92.6† | 91.2† | 80.6† | 84.5 | 88.1 |
| Llama3-70B | 93.7† | 97.7† | 91.7† | 83.2† | 89.5 | 79.8† | 93.4† | 91.6† | 78.9† | 84.1 | 88.4 |
| Qwen2.5-72B | 95.5† | 98.8† | 91.9† | 89.7† | 97.5† | 85.3† | 97.4† | 94.0† | 82.2† | 84.3† | 91.7 |

Table 13: Extended reproducible performance scores across models and tasks using OLMES, providing robust, meaningful comparisons across a wide range of models and tasks. † indicates use of the MCF score.

| Model↓ | ARC-Challenge Evaluations: Ref1 | Ref2 | Ref3 | Ref4 | Ref5 | Ref6 | OLMES | OpenbookQA Evaluations: Ref2 | Ref4 | Ref5 | Ref7 | Ref8 | OLMES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MPT-7B | 47.7 | 42.6 | | | 46.5 | | 45.7 | 51.4 | | 48.6 | | | 52.4 |
| RPJ-INCITE-7B | 46.3 | | | | 42.8 | | 45.3 | | | 49.4 | | | 49.0 |
| Falcon-7B | 47.9 | 42.4 | | 44.5 | 47.5 | | 49.7 | 51.6 | 44.6 | 53.0 | | 26.0† | 55.2 |
| Mistral-7B | 60.0 | | 55.5 | 54.9 | | | 78.6† | | | | 52.2 | 77.6† | 80.6† |
| Llama2-7B | 53.1 | 45.9 | 43.2 | 45.9 | 48.5 | 53.7† | 54.2 | 58.6 | 58.6 | 48.4 | 58.6 | 54.4† | 57.8 |
| Llama2-13B | 59.4 | 49.4 | 48.8 | 49.4 | | 67.6† | 67.3† | 57.0 | 57.0 | | 57.0 | 63.4† | 65.4† |
| Llama3-8B | 60.2 | | | | | 78.6† | 79.3† | | | | | 76.6† | 77.2† |
| Num shots | 25 | 0 | 0 | 0 | 0 | 25 | 5 | 0 | 0 | 0 | 0 | 5 | 5 |
| Curated shots | No | | | | | No | Yes | | | | | No | Yes |
| Formulation | CF | CF | CF? | CF | CF | MCF | MCF/CF | CF | CF | CF | CF | MCF | MCF/CF |
| Normalization | char | char | ? | char? | pmi | none | none/pmi | pmi | pmi? | pmi | pmi? | none | none/pmi |

| Ref | Reference citation | Ref | Reference citation |
|---|---|---|---|
| **Ref1** | HF Open LLM Leaderboard (Beeching et al., 2023) | **Ref5** | OLMo paper (Groeneveld et al., 2024) |
| **Ref2** | Llama2 paper (Touvron et al., 2023a) | **Ref6** | Llama3 model card (AI@Meta, 2024) |
| **Ref3** | Mistral 7B (Jiang et al., 2023) | **Ref7** | Gemma paper (Gemma Team et al., 2024) |
| **Ref4** | Falcon paper (Almazrouei et al., 2023) | **Ref8** | HELM Lite Leaderboard (Liang et al., 2023) |

Table 14: Extended version of Table 1 showing scores reported in different references for LLM performances on ARC-Challenge and OpenbookQA. Scores indicated with † are using multiple-choice formulation (MCF) rather than "cloze" formulation (CF) (see Section 2.1 for definitions). Entries with "?" denote either undocumented or mixed approaches across models. Different references use different evaluation setups, some of which are not fully specified, so conclusions about which models perform best are not reproducible.

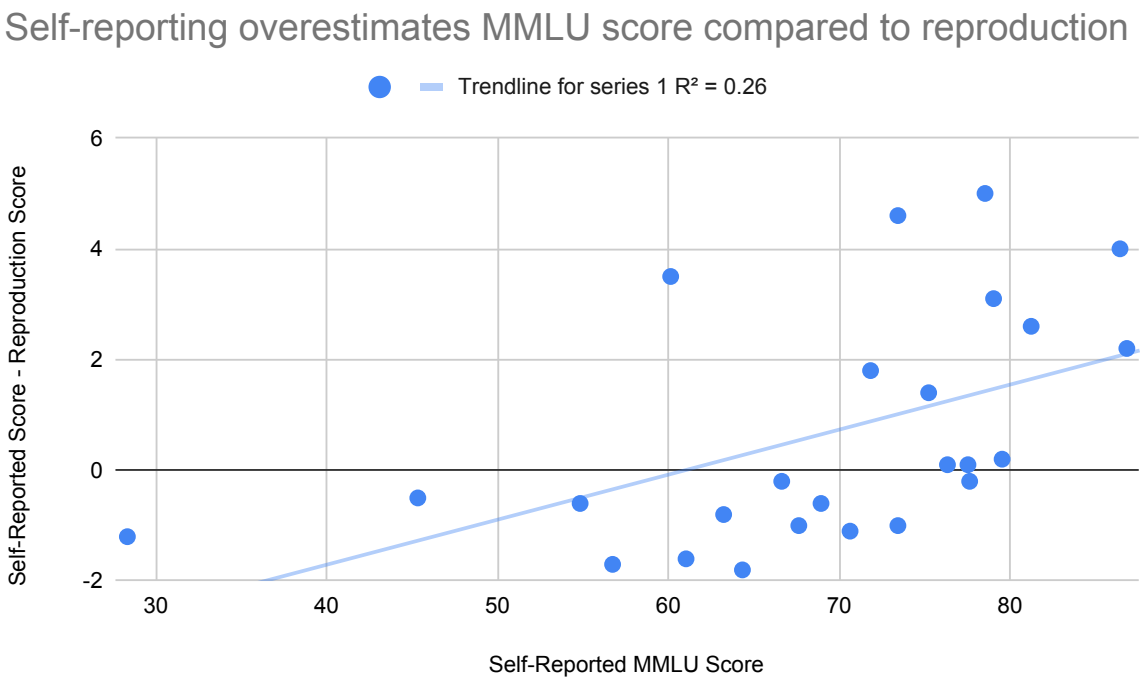


Figure 4: Self-reporting overestimates MMLU score compared to reproduction, from https://crfm.stanford.edu/2024/05/01/helm-mmlu.html. Each point corresponds to a model, the x-axis shows self-reported MMLU score, and the y-axis shows the difference between the self-reported score and the reproduced score. Points above the y=0 line have higher self-reported performance than the reproduction; the trend line has a positive slope, indicating that on average, the higher the self-reported score the more they overestimate performance compared to the reproduction.

| | |
|---|---|
| Prompt | Question: George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?<br>Answer: dry palms<br><br>Question: Which of the following statements best explains why magnets usually stick to a refrigerator door?<br>Answer: The refrigerator door contains iron.<br><br>Question: A fold observed in layers of sedimentary rock most likely resulted from the<br>Answer: converging of crustal plates.<br><br>Question: Which of these do scientists offer as the most recent explanation as to why many plants and animals died out at the end of the Mesozoic era?<br>Answer: impact of an asteroid created dust that blocked the sunlight<br><br>Question: Which of the following is a trait that a dog does NOT inherit from its parents?<br>Answer: the size of its appetite<br><br>Question: A boat is acted on by a river current flowing north and by wind blowing on its sails. The boat travels northeast. In which direction is the wind most likely applying force to the sails of the boat?<br>Answer: |
| Completion | east |

Figure 5: OLMES 5-shot prompt example for ARC-CHALLENGE (CF).

- Skip ambiguous instances
- Skip instances that hint at discrimination or otherwise deemed inappropriate
- Skip instances if the same label has appeared frequently (e.g., 4 consecutive instances with gold label 'C', keep better ones out of those)
- If instances are grouped/labeled by topic, choose instances to be diverse (e.g., first 3 are all about a certain topic, pick from later ones to ensure diversity).
- If you end up with less than 7 instances that cover the label space or range of different topics, look at the next batch of 10.
- Finally, reorder instances to obtain a somewhat balanced output of answer labels – the first 5 shots should cover the space of answer labels.

Note that a few more than 5 shots per dataset were curated in the process, though in practice we are just using the first 5.

## H OLMES prompt formats for each task

In Figure 5 we show an example of a full 5-shot prompt from ARC-CHALLENGE (CF). Then we show single instance formatting for each of the 10 tasks in Figures 6- 25. For each task, we show both the MCF and CF formats.

All curated few-shot examples and prompt formatting code are available by accessing https://github.com/allenai/olmes.

| | |
|---|---|
| Prompt | Question: George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?<br>A. dry palms<br>B. wet palms<br>C. palms covered with oil<br>D. palms covered with lotion<br>Answer: |
| Completion | A |

Figure 6: OLMES prompt example for ARC-CHALLENGE (MCF).

| | |
|---|---|
| Prompt | Question: George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?<br>Answer: |
| Completion | dry palms |

Figure 7: OLMES prompt example for ARC-CHALLENGE (CF).

| | |
|---|---|
| Prompt | Question: Lichens are symbiotic organisms made of green algae and fungi. What do the green algae supply to the fungi in this symbiotic relationship?<br>A. carbon dioxide<br>B. food<br>C. protection<br>D. water<br>Answer: |
| Completion | B |

Figure 8: OLMES prompt example for ARC-EASY (MCF).

| | |
|---|---|
| Prompt | Question: Lichens are symbiotic organisms made of green algae and fungi. What do the green algae supply to the fungi in this symbiotic relationship?<br>Answer: |
| Completion | food |

Figure 9: OLMES prompt example for ARC-EASY (CF).

| | |
|---|---|
| Prompt | Persian language – Persian, also known by its endonym Farsi, is one of the Western Iranian languages within the Indo-Iranian branch of the Indo-European language family. It is primarily spoken in Iran, Afghanistan (officially known as Dari since 1958), and Tajikistan (officially known as Tajiki since the Soviet era), and some other regions which historically were Persianate societies and considered part of Greater Iran. It is written in the Persian alphabet, a modified variant of the Arabic script, which itself evolved from the Aramaic alphabet.<br>Question: do iran and afghanistan speak the same language?<br>A. yes<br>B. no<br>Answer: |
| Completion | A |

Figure 10: OLMES prompt example for BOOLQ (MCF).

| | |
|---|---|
| Prompt | Persian language – Persian, also known by its endonym Farsi, is one of the Western Iranian languages within the Indo-Iranian branch of the Indo-European language family. It is primarily spoken in Iran, Afghanistan (officially known as Dari since 1958), and Tajikistan (officially known as Tajiki since the Soviet era), and some other regions which historically were Persianate societies and considered part of Greater Iran. It is written in the Persian alphabet, a modified variant of the Arabic script, which itself evolved from the Aramaic alphabet.<br>Question: do iran and afghanistan speak the same language?<br>Answer: |
| Completion | yes |

Figure 11: OLMES prompt example for BOOLQ (CF).

| | |
|---|---|
| Prompt | Question: Sammy wanted to go to where the people were. Where might he go?<br>A. race track<br>B. populated areas<br>C. the desert<br>D. apartment<br>E. roadblock<br>Answer: |
| Completion | B |

Figure 12: OLMES prompt example for COMMONSENSEQA (MCF).

| | |
|---|---|
| Prompt | Question: Sammy wanted to go to where the people were. Where might he go?<br>Answer: |
| Completion | populated areas |

Figure 13: OLMES prompt example for COMMONSENSEQA (CF).

| | |
|---|---|
| Prompt | Health: How to cope with suicidal thoughts. Put off any plans. Promise yourself that you'll wait 48 hours before doing anything. Remember, thoughts don't have the power to force you to act.<br>Choose the best continuation:<br>A. Even when you do, there may be a small image of the future still lurking around your brain. For instance, don't tell yourself that you can't make it.<br>B. You're doing something, and no one can force you to act. It's completely natural to feel negative thoughts before you act.<br>C. Do not panic if people talk to you (even if it's about quitting smoking). Have a plan for how you're going to react to a group of people who bring on suicidal thoughts.<br>D. Sometimes extreme pain can distort our perception. Waiting before taking action will give your mind time to clear.<br>Answer: |
| Completion | D |

Figure 14: OLMES prompt example for HELLASWAG (MCF).

| | |
|---|---|
| Prompt | Health: How to cope with suicidal thoughts. Put off any plans. Promise yourself that you'll wait 48 hours before doing anything. Remember, thoughts don't have the power to force you to act. |
| Completion | Sometimes extreme pain can distort our perception. Waiting before taking action will give your mind time to clear. |

Figure 15: OLMES prompt example for HELLASWAG (CF).

| | |
|---|---|
| Instruction Prompt | The following are multiple choice questions (with answers) about abstract algebra.<br>Question: Find all c in Z_3 such that Z_3[x]/(x^2 + c) is a field.<br>A. 0<br>B. 1<br>C. 2<br>D. 3<br>Answer: |
| Completion | B |

Figure 16: OLMES prompt example for MMLU (abstract_algebra) (MCF).

| | |
|---|---|
| Instruction Prompt | The following are multiple choice questions (with answers) about abstract algebra.<br>Question: Find all c in Z_3 such that Z_3[x]/(x^2 + c) is a field.<br>Answer: |
| Completion | 1 |

Figure 17: OLMES prompt example for MMLU (abstract_algebra) (CF).

| | |
|---|---|
| Prompt | Question: When standing miles away from Mount Rushmore<br>A. the mountains seem very close<br>B. the mountains are boring<br>C. the mountains look the same as from up close<br>D. the mountains seem smaller than in photographs<br>Answer: |
| Completion | D |

Figure 18: OLMES prompt example for OpenbookQA (MCF).

| | |
|---|---|
| Prompt | Question: When standing miles away from Mount Rushmore<br>Answer: |
| Completion | the mountains seem smaller than in photographs |

Figure 19: OLMES prompt example for OpenbookQA (CF).

| | |
|---|---|
| Prompt | Goal: how do you stab something?<br>A. stick a sharp object through it.<br>B. pin it with a sharp object.<br>Answer: |
| Completion | A |

Figure 20: OLMES prompt example for Physical Interaction QA (MCF).

| | |
|---|---|
| Prompt | Goal: how do you stab something?<br>Answer: |
| Completion | stick a sharp object through it. |

Figure 21: OLMES prompt example for Physical Interaction QA (CF).

| | |
|---|---|
| Prompt | Question: Cameron decided to have a barbecue and gathered her friends together. How would Others feel as a result?<br>A. like attending<br>B. like staying home<br>C. a good friend to have<br>Answer: |
| Completion | A |

Figure 22: OLMES prompt example for Social IQa (MCF).

| | |
|---|---|
| Prompt | Question: Cameron decided to have a barbecue and gathered her friends together. How would Others feel as a result?<br>Answer: |
| Completion | like attending |

Figure 23: OLMES prompt example for SOCIAL IQA (CF).

| | |
|---|---|
| Prompt | Fill in the blank: John moved the couch from the garage to the backyard to create space. The ___ is small.<br>A. garage<br>B. backyard<br>Answer: |
| Completion | A |

Figure 24: OLMES prompt example for WINOGRANDE (MCF).

| | |
|---|---|
| Prompt1 | John moved the couch from the garage to the backyard to create space. The garage |
| Prompt2 | John moved the couch from the garage to the backyard to create space. The backyard |
| Completion | is small. |

Figure 25: OLMES prompt example for WINOGRANDE (CF). In this case the completions are the same for each answer choice, but the prompt is different.