GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

HPS

Seminar Report

LLM Benchmarking Frameworks and their limitations

Niclas Unger

MatrNr: 21880008

Supervisor: Sadegh Keshtkar

Georg-August-Universität Göttingen
Institute of Computer Science

March 31, 2026

Abstract

Large language models have rapidly become central tools in modern natural language processing, enabling applications ranging from conversational chatting agents to complex reasoning. However, a reliable evaluation and comparison between these models remains a challenge, since their outputs are sensitive to context, prompting and evaluation methodology. This can lead to misleading benchmarking results. Existing approaches still only apply patchwork like fixes for these limitations by utilizing standardized benchmark datasets and evaluation frameworks, such MMLU, GSM8K and EleutherAI's Language Model Evaluation Harness. However, these methods are often still affected by the same limitations, such as data contamination, prompt sensitivity and answer extraction techniques. This report combines a structured literature review with small-scale experiments on different prompting configurations, giving an initial insight into the topic and showing that evaluation choices, especially prompt formatting and output extraction, can significantly alter perceived model performance.

General structure of an abstract, write 1 to 2 sentences per section

1. A general statement introducing the broad research area of the particular topic being investigated.
2. An explanation of the specific problem (difficulty, obstacle, challenge) to be solved.
3. A review of existing or standard solutions to this problem and their limitations.
4. An outline of the proposed new solution.
5. A summary of how the solution was evaluated and what the outcomes of the evaluation were.

You may find the following resources useful:

- `https://www.grammarly.com/blog/write-an-abstract/`
- `https://www.editage.com/insights/manuscript-structure-how-to-convey-your-most-im`
- More useful links: `https://hps.vi4io.org/teaching/ressources/start`

Declaration on the use of ChatGPT and comparable tools in the context of examinations

In this work I have used ChatGPT or another AI as follows:

- ☐ Not at all
- ☑ During brainstorming
- ☐ When creating the outline
- ☐ To write individual passages, altogether to the extent of 0% of the entire text
- ☐ For the development of software source texts
- ☑ For optimizing or restructuring software source texts
- ☑ For proofreading or optimizing
- ☑ Further, namely: Help with LaTeX syntax and commands

I hereby declare that I have stated all uses completely.
Missing or incorrect information will be considered as an attempt to cheat.

Contents

List of Tables

List of Figures

List of Listings

List of Abbreviations

LLM Large Language Model

API Application Programming Interface

HELM Holistic Evaluation of Language Models

MMLU Massive Multitask Language Understanding

MMMLU Multilanguage Massive Multitask Language Understanding

Lm Eval Harness Language Model Evaluation Harness

SOTA state of the art

1 Introduction

Large language models (LLMs) have rapidly become a central part of not just natural language processing research, but of academia and information sourcing in general. They enable applications ranging from conversational chatting agents to code generation and complex reasoning. As an increasing variety of models is deployed in real-world settings, with models like GPT, Clause or Gemini competing for the top, we need reliable evaluation methods to compare models. LLM benchmarks are commonly used to measure and compare model capabilities, as well as to track progress from one version to the next. This is supposed to allow for a realistic comparison between models and a fair competition between developers, as benchmarking scores are the main indicator of progress of models.

However, evaluating LLMs is an inherently challenging task. Unlike with many traditional machine learning tasks with clearly defined ground truth labels, boxes or classes, LLM outputs are often complex, convoluted, dependent on context and sensitive to prompting and evaluation methods. Factors such as these make it difficult to ensure that a benchmark result fairly and accurately reflects a models capabilities rather than just its compatibility or compliance with the specific data or evaluation implementation.

To address this problem, a variety of benchmarking datasets and benchmark evaluation frameworks have been proposed over the last years. These include code repositories for standardized and systematic execution of benchmarks on LLMs, as well as benchmarks that themselves seek to mitigate some of the complications inherent to them. While these approaches provide useful starting points, they still suffer from several limitations, such as data contamination, prompt sensitivity, metric sensitivity, correctness and, with stronger and stronger models, changing requirements to difficulty.

This report investigates existing LLM benchmarks and benchmarking frameworks and analyzes their design and inherent limitations. In particular, we examine well-known benchmarks and their evaluation setups, highlighting their strengths and weaknesses. We also perform some small scale experiments, using EleutherAI's Language Model Evaluation Harness, to illustrate some of these weaknesses and how severely they can affect evaluation. Based on this, we argue that currently, benchmarking practices are still insufficiently developed for fully and reliably capturing LLM capabilities, and that further research into this field is still very much required.

1.1 Contributions

The main contributions of this report are as follows:

1. Consolidated overview of modern LLM benchmarking practices and methods.
2. Analysis of key relevant benchmarking limitations, with a focus on data contamination, prompt sensitivity and output handling.
3. An empirical investigation of prompt sensitivity and the significance of answer extraction methods during the evaluation process, using GSM8K and EleutherAI's LM Eval Harness.

1.2 Outline of the Report

The remainder of this report is structured as follows. Section two provides a background on LLM benchmarks and benchmarking frameworks and their roles in standardizing LLM evaluation procedures. Section 3 describes the methodology used for this report, the including literature review and an illustrative experiment. Section 4 describes two specific limitations, data contamination and prompt sensitivity and detail with the help of two relevant benchmarks. Section 5 discusses the results of the experiments on prompt sensitivity and the significance of evaluation processes. Section 7 concludes the report.

2 Background

2.1 LLM Benchmarks

In the context of large language models, benchmarks are evaluation setups used to measure model performance. A benchmark typically consists of a fixed set of questions, each belonging to one or many different subjects. Each question also comes with a ground truth answer for evaluation. Furthermore, every benchmark is equipped with some kind of performance evaluation metric by which the model answers can be scored. Such metrics include, for example, accuracy, efficiency, robustness, bias or toxicity and many custom metrics. Benchmarks can be grouped according to the type of capability that they measure and evaluate. *Knowledge-based* benchmarks test a model's ability to recall factual information. *Reasoning-based* benchmarks test logical thinking and step-by-step reasoning capabilities. Additionally, there are also benchmarks testing *language* capabilities of a model, such as the ability to translate text. Other benchmarks evaluate aspects such as *bias*, *toxicity* or *safety* of a model's responses. Finally, specifically relevant for computer scientists, are *coding* based benchmarks, which judge a model's ability to understand, improve or generate code.

Readers will frequently encounter the term *tasks* when researching this topic. However, a task is not uniformly defined in literature. It can be a type of question, like multiple choice, code generation or arithmetic, but different papers use the term to describe slightly different things. For instance, the authors of Hendrycks et al. [Hen+21a] use the term to describe the different subjects in their benchmark, such as sociology, philosophy or astronomy.

2.1.1 Representative benchmarks

One of the most well-known benchmarks is MMLU, which stands for Measuring Massive Multitask Language Understanding. It is a dataset consisting of around 16000 multiple choice questions, spanning across 57 tasks/subjects. The questions are split into a small developer, validation and test set. The developer set contains 5 questions with answers per subject, which are used for the few-shot prompting. The validation set can be used to adjust hyperparameters and consists of 1540 questions. The rest of the questions are part of the test set and are used for the final evaluation only. [Hen+21a]

To get an understanding of how some of these questions might look like, here is an example from the *conceptual physics* subset of questions:

Example Question from MMLU (Figure 4 in [Hen+21a]): When you

drop a ball from rest it accelerates downward at $9.8\,\mathrm{m/s^2}$. If you instead throw it downward assuming no air resistance, its acceleration immediately after leaving your hand is

(A)	$9.8\,\mathrm{m/s^2}$	✓
(B)	more than $9.8\,\mathrm{m/s^2}$	X
(C)	less than $9.8\,\mathrm{m/s^2}$	X
(D)	Cannot say unless the speed of throw is given	X

MMLU is scored via accuracy. To evaluate a model's response, one has to compute the log-likelihoods of the model to generate each answer, for example "$9.8\,\mathrm{m/s^2}$", as a continuation to the prompt. The option with the highest likelihood to be generated is chosen as the model's answer and compared to the ground truth for each question.

Other notable examples for well-known benchmarks include: The math-focused GSM8K [Cob+21] and MATH [Hen+21b], the knowledge-reasoning-focused ARC [Cla+18] and the common sense-reasoning-focused CommonsenseQA [Tal+19] and HellaSwag [Zel+19]. Furthermore, several overhauled versions of existing benchmarks have been proposed, such as MMLU-Pro [Wan+24], MMMLU [OF24] or MMLU-Redux [Gem+25]. MMLU-Pro adds more challenging questions, MMMLU extends the benchmark into other languages, while MMLU-Redux strives to correct incorrect or unclear questions.

2.2 Benchmarking frameworks

A benchmarking framework can be defined as a (software) system that standardizes and/or automates the evaluation of benchmarks on LLMs. A framework will usually allow for the execution of one or multiple benchmarks on one or multiple models. They define evaluation protocols, like prompt templates (zero-shot, few-shot, chain-of-thought), decoding settings (temperature, top_p, etc.) and output scoring. An important aspect is that frameworks ensure comparability and reproducibility across models and runs. Depending on the framework, they might also support scalable execution option, such as batching or parallelism.

A widely used framework, which we also utilize in our experiments, is EleutherAI's Language Model Evaluation Harness. The authors describe it as a "Unified framework to test generative language models on a large number of different evaluation tasks". It supports more than 60 benchmarks and many different LLMs via compatibility with many models available on HuggingFace. It is also compatible with commercial APIs, like the OpenAI API or in principal any API that is using the OpenAI API python package. LM Eval Harness further supports local models and custom benchmarks, as well as custom prompts and evaluation metrics. Some forms GPU parallelism are also supported. [Gao+21]

Other notable examples of benchmarking frameworks include OpenCompass [Ope23], HELM [Lia+23] and PromptBench [Pro24] and ChatbotArena [Org23].

3 Methodology

The goal for this report was to get a good understanding of what benchmarks and benchmarking frameworks are, how they function, what limitations appear, how relevant these limitations are and how researches have tried to combat these limitations. To do that,

we have done a thorough search through relevant literature, especially papers on well-known benchmarks. We have, for some of these benchmarks, extracted their unique ideas to present them in a consolidated form here. From among the existing limitations, we have chosen to focus on data contamination and prompt sensitivity. Then, to illustrate ourselves, the effects of prompt sensitivity on the evaluation results of benchmarks, we have conducted some small experiments on LLMs, using GSM8K via a benchmarking framework, and utilizing varying prompting templates.

In the experiments, we used EleutherAI's benchmarking framework Language Model Evaluation Harness (LM Eval Harness). Specifically, we investigated the effects of different prompting formats on the evaluation results of LLMs under GSM8K. When using GSM8K via LM Eval Harness, the framework wraps each question into a "Question:{question}\nAnswer:" template. Since the modification of the dataset GSM8K itself is much more time consuming and difficult to do uniformly, we have opted to instead modify this prompt wrapping template. Excluding the default template, we have tested GSM8K for 6 different custom prompting templates 4. Due to limited access to the GWDG ChatAI API, we restricted our experiments to 50 questions per run. In addition, to further reduce the API rate load, we opted to use a 1-shot format (instead of the default of 5), meaning every question given to the model is preceded by one example question in the same format, but including the correct answer. Reducing the number of few-shot examples per prompt does noticeably decrease model performance, but still allows for a good comparison between prompt templates. Additionally, we also compared these experiments to a 0-shot approach. To get a comparison between both a small and a larger model, we chose meta-llama-3.1-8b-instruct and devstral-2-123b-instruct-2512 from among the models available via the GWDG ChatAI platform.

The following code illustrates the core functionality of the experiments.

```python
from lm_eval import simple_evaluate
from openai import OpenAI

results = simple_evaluate(
    model="openai-completions",
    model_args={
        "model": "meta-llama-3.1-8b-instruct",
        "base_url": "https://chat-ai.academiccloud.de/v1/completions",
        "apply_chat_template": True,
        "num_concurrent": 1,
    },
    tasks=["gsm8k_default"],
    limit=50,
    num_fewshot=1
    )
```

Listing 1: API call example for evaluation of GSM8K on Llama 3.1, using LM Eval Harness and the OpenAI API package

As illustrated in Listing 1, LM Eval Harness allows for many customization options. The given code snippet evaluates the first 50 questions of the GSM8K dataset on the

Llama 3.1 model, as described above. It uses the OpenAI API package and the provided url to access the model on the GWDG ChatAI platform. A corresponding API key is required to run the evaluation. Batching is possible by varying *num_ concurrent*, however increasing this number was not feasible in our use case due to API rate limit restrictions. Applying a chat template adds a further wrapper around the prompt, which is highly recommended to retain performance for any instruct models.

4 Limitations

Benchmarks are affected by several, sometimes closely related, limitations. A significant problem for benchmarks is data contamination. Data contamination happens, when an LLM sees the benchmark or similar data during training and therefore gains a performance boost on the benchmark not because the model is capable, but because it already saw the data during training. Since benchmarks are public and static datasets, this is a real danger, especially for models that are trained on large amounts of data that is available on the internet. Further, this implies that benchmark scores of newer models on older benchmarks might be inflated, compared to previous models' scores. [DMV24]

Another limitation of benchmarks is prompt sensitivity. A models response can vary wildly, based on how a question or prompt is formulated. Some models might respond better to simple wording, while others might respond better to questions that were formulated similar to their training data (e.g. instruct models). This means that the choice of wording in benchmark questions is highly relevant to ensure fair and representative benchmarking scores.

The third limitation lies in the evaluation process. For every benchmark evaluation, it has to be decided how a model's outputs are to be processed and scored. Even for accuracy-based benchmarks, it is often unclear, how a model response should be interpreted. As can be seen during our experiments (2), the two different approaches to extract the model's answer often lead to vastly different scores. One solution is to use log-likelihoods like, for example, in the MMLU paper [Hen+21a], but it is not 100% clear, how reliable this approach is. Another factor to consider is metric sensitivity. Authors of benchmarks such as [Zhu+24] often define their of evaluation metric. These metrics can introduce some amount of ambiguity or obscurity into the evaluation process. In general, one can reason, that just like in other machine learning tasks, there are multiple possible metrics that each have their upsides and downsides.

Another limiting factor of benchmarks is the increasing capability of modern models. Benchmarks that might have been adequate to compare state-of-the-art (SOTA) models five years ago might only yield negligible differences in scores between today's SOTA models. Changing capabilities of LLMs mean that benchmarks quickly become outdated.[Akh+26]

The final limitation that we will discuss here is correctness. As is evident by overhauled benchmark versions like [Gem+25], the issue with generating such large datasets of questions (which are often machine generated, see e.g. [Zel+19]) is that it is difficult to ensure that every question is clear and has an unambiguous, correct answer. This issue further decreases the validity of many large benchmarks.

5 Experimental Results

Model	Prompting Template	Few-Shot strict	Few-Shot flexible	0-Shot strict	0-Shot flexible
meta-llama-3.1-8b-instruct	default (3-run average)	28%	64%	0%	30%
	distraction	34%	2%	0%	10%
	simple	34%	12%	0%	22%
	strong wording	26%	60%	0%	32%
	expert	6%	8%	0%	18%
devstral-2-123b-instruct-2512	default	88%	90%	0%	42%
	distraction	76%	6%	0%	46%
	simple	84%	6%	0%	12%
	strong wording	80%	88%	0%	48%
	expert	90%	8%	0%	42%

Table 1: Partial results (accuracy in percent) for GSM8K under different prompting templates using LM Eval Harness.

Table 1 shows the main results from our experiments. It demonstrates that varying prompt templates, even without changing the benchmark question itself, can have significant effect on evaluation scores. *Strict* stands for strict match and represents the calculated accuracy, when applying the following regular expression:

```
#### (\-?[0-9\.\,]+)
```

This expression indicates that the framework looks for four hashtags, followed by a number and this number is then assumed to the model's response. Since this format with a number followed after hashtags has to be copied from the few-shot examples given before each question, it is obvious, why the 0-shot experiments always yield 0% accuracy, since the model doesn't know the correct answer format. *Flexible* stands for flexible extract and means that the framework simply chooses the first number appearing in the model's output as the model's response. Upon manual review of the results, it can be observed that whenever the flexible extract accuracy is lower than the strict extract accuracy, it appears to always be an issue of the way that the model answer is extracted from the response, not of the model response itself. Similarly, the reason why most strict extract scores are low is because the model answer doesn't follow the correct response format demonstrated during the few-shot process, which leads to the strict extract method failing (returns 0 "invalid").

We can learn from these results that prompt sensitivity and answer extraction are sometimes two tightly connected problems. The results also shows a significant effect of prompt format and especially answer extraction methodology on a model's benchmarking score. It is also evident that the large Devstral model is overall more resistant to changes in prompting format, especially with the strict match answer extraction method. Further it is apparent that 0-shot evaluations, at least for GSM8K, do often not yield useful benchmarking results.

6 Conclusion

In this report, we have examined the current state of large language model benchmarking with a focus on practical limitations and the methodological designs. Through a literature review of widely used benchmarks and frameworks and with targeted experiments using GSM8K via the LM Evaluation Harness, we have shown that benchmarking scores are far from being a reliable, model-intrinsic representation of a models capabilities.

The main finding is that evaluation outcomes are highly sensitive to specific implementation details. In particular, prompt formulation and wrapping and answer extraction strategies play a large role in the measured benchmark performance without changing anything in the underlying model. The experimental results demonstrate that even supposedly small variations in prompting templates or output handling can lead to large changes in accuracy, leading to untrustworthy comparisons between models. This highlights that reported benchmark scores often reflect a combination of model ability, prompt design, and evaluation methodology rather than just pure model quality alone.

These findings suggest that current benchmarking practices have not yet overcome the inherent limitations of benchmarks. While existing benchmarks still present useful tools for comparison of models, results must be interprented with caution.

References

[Akh+26] Mubashara Akhtar et al. *When AI Benchmarks Plateau: A Systematic Study of Benchmark Saturation*. 2026. arXiv: 2602.16763 [cs.AI]. URL: https://arxiv.org/abs/2602.16763.

[Cla+18] Peter Clark et al. *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*. 2018. arXiv: 1803.05457 [cs.AI]. URL: https://arxiv.org/abs/1803.05457.

[Cob+21] Karl Cobbe et al. *Training Verifiers to Solve Math Word Problems*. 2021. arXiv: 2110.14168 [cs.LG]. URL: https://arxiv.org/abs/2110.14168.

[DMV24] Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. *ConStat: Performance-Based Contamination Detection in Large Language Models*. 2024. arXiv: 2405.16281 [cs.CL]. URL: https://arxiv.org/abs/2405.16281.

[Gao+21] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.0.1. Sept. 2021. DOI: 10.5281/zenodo.5371628. URL: https://doi.org/10.5281/zenodo.5371628.

[Gem+25] Aryo Pradipta Gema et al. *Are We Done with MMLU?* 2025. arXiv: 2406.04127 [cs.CL]. URL: https://arxiv.org/abs/2406.04127.

[Hen+21a] Dan Hendrycks et al. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: https://arxiv.org/abs/2009.03300.

[Hen+21b] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG]. URL: https://arxiv.org/abs/2103.03874.

[Lia+23] Percy Liang et al. *Holistic Evaluation of Language Models*. 2023. arXiv: 2211.09110 [cs.CL]. URL: https://arxiv.org/abs/2211.09110.

[OF24] OpenAI and Hugging Face. *MMMLU: Multilingual Massive Multitask Language Understanding Dataset*. https://huggingface.co/datasets/openai/MMMLU. Accessed: 2026-03. 2024. URL: https://huggingface.co/datasets/openai/MMMLU.

[Ope23] OpenCompass Team. *OpenCompass: A Platform for Large Language Model Evaluation*. https://github.com/open-compass/opencompass. GitHub repository. 2023.

[Org23] LMSYS Org. *Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference*. https://chat.lmsys.org/. 2023.

[Pro24] PromptBench Contributors. *PromptBench: A Unified Library for Evaluating and Understanding Large Language Models*. https://github.com/microsoft/promptbench. Last accessed: 2026-02-04. 2024.

[Tal+19] Alon Talmor et al. *CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge*. 2019. arXiv: 1811.00937 [cs.CL]. URL: https://arxiv.org/abs/1811.00937.

[Wan+24] Yubo Wang et al. *MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark*. 2024. arXiv: 2406.01574 [cs.CL]. URL: https://arxiv.org/abs/2406.01574.

[Zel+19] Rowan Zellers et al. *HellaSwag: Can a Machine Really Finish Your Sentence?* 2019. arXiv: 1905.07830 [cs.CL]. URL: https://arxiv.org/abs/1905.07830.

[Zhu+24] Jingming Zhuo et al. *ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs.* 2024. arXiv: 2410.12405 [cs.CL]. URL: https://arxiv.org/abs/2410.12405.

A Tables

Model	Prompting Template	Few-Shot strict	Few-Shot flexible	0-Shot strict	0-Shot flexible
meta-llama-3.1-8b-instruct	default (1st run)	30%	60%	0%	30%
	default (2nd run)	24%	66%	/	/
	default (3rd run)	30%	68%	/	/
	default (3-run average)	28%	64%	0%	30%
	force reasoning	32%	56%	0%	24%
	distraction	34%	2%	0%	10%
	simple	34%	12%	0%	22%
	strong wording	26%	60%	0%	32%
	expert	6%	8%	0%	18%
	exact	26%	4%	/	/
devstral-2-123b-instruct-2512	default	88%	90%	0%	42%
	distraction	76%	6%	0%	46%
	simple	84%	6%	0%	12%
	strong wording	80%	88%	0%	48%
	expert	90%	8%	0%	42%

Table 2: Full results for first 50 questions of GSM8K under different prompting templates using Lm Eval Harness.

Model	Prompting Template	5-Shot strict	5-Shot flexible
meta-llama-3.1-8b-instruct	default	60%	74%
devstral-2-123b-instruct-2512	default	88%	90%

Table 3: 5-shot results for first 50 questions of GSM8K using Lm Eval Harness.

Furthermore, performing 150 questions of the GSM8K benchmark, instead of just 50, on the Llama model yielded the following results (using the default prompt template with few-shot): 32% accuracy for strict match and 75.3% accuracy for flexible match.

Template	**Prompt Wrapper**
default	Question: {{question}}\nAnswer:
distraction	The Eiffel tower is about 300m tall. {{question}}\nAnswer:
exact	Answer the following question by only giving the final answer: {{question}}\nAnswer:
expert	As an AI expert in math, please provide an answer to the following question: {{question}}\nAnswer:
force reasoning	Solve the following question step by step. Question: {{question}}\nAnswer:
simple	{{question}}
strong wording	You have to answer this question correctly! Question: {{question}}\nAnswer:

Table 4: Prompt wrapping templates used in the GSM8K experiments