Benchmarking Large Language Models: Opportunities and Challenges

Miro Hodak[1]*, David Ellison[2], Chris Van Buren[2], Xiaotong Jiang[2], Ajay Dholakia[2]
[1]AMD, Data Center Solutions Group, Austin, TX, USA
[2]Lenovo, Infrastructure Solutions Group, Morrisville, NC, USA
[1]Miro.Hodak@amd.com
[2]{dellison, cvanburen, jiangxt, adholakia}@lenovo.com
*Corresponding author

Abstract. With exponentially growing popularity of Large Language Models (LLMs) and LLM-based applications like ChatGPT and Bard, the Artificial Intelligence (AI) community of developers and users are in need of representative benchmarks to enable careful comparison across a variety of use cases. The set of metrics has grown beyond accuracy and throughput to include energy efficiency, bias, trust and sustainability. This paper aims to provide an overview of popular LLMs from a benchmarking perspective. Key LLMs are described, and the associated datasets are characterized. A detailed discussion of benchmarking metrics covering training and inference stages is provided and challenges in evaluating these metrics are highlighted. A review of recent performance and benchmark submissions is included, and emerging trends are summarized. The paper lays the foundation for developing new benchmarks to allow informed comparison of different AI systems based on combinations of models, datasets, and metrics.

Keywords: Artificial Intelligence, Inference, Training, MLPerf, TPCx-AI, Deep Learning, Performance, Large Language Models

1 Introduction

Currently, Large Language Models (LLMs) are the hottest trend in Artificial Intelligence (AI) development and in datacenter workloads in general. Enabled by the development of transformers, they have replaced vision workloads as the cutting edge of AI. While LLMs have been popular in the AI community at least since the release of BERT in 2018, it was public release of ChatGPT in November 2022 that captured public attention and became a watershed moment in LLMs. ChatGPT demonstrated that it can answer questions, write text and computer programs, pass exams, and much more. It became the fastest service to reach 100 million users [1].

The success of ChatGPT has catalyzed LLM development and deployment. It has led to multiple companies rushing to deploy their own LLM models. For example, in February 2023, Google announced its own rival chatbot, Bard [2].

At the same time, it has become evident that LLM training and deployment is very costly. While GPT-3-based ChatGPT remains free, the large size of the models combined with the need for large throughput - i.e., generating a large amount of text quickly - means that deployments usually require a large number of accelerators. This makes benchmarking indispensable for evaluating different modes of deployments. Yet, LLM benchmarking lags far behind deployments.

Beyond the rapid public adoption, other reasons benchmarking has lagged are a lack of standardization and high compute requirements. So far, the gap has been filled with many ad hoc efforts with chip makers claiming leadership in workloads best suited to their products and individuals and enthusiasts publishing their own results. Given the importance of this area, it is imperative that standardization and benchmarking quickly catch up. On the standardization side, there is a need for identifying the most representative LLMs along with datasets and with accuracy criteria – unlike in vision workloads, accuracy scoring in LLMs is less obvious and several competing metrics have been developed. Once the models and datasets are identified, a set of benchmarking rules can be defined to arrive at a fair LLM evaluation.

The current state of the field is that only a single standards organization has brought LLM benchmarks: MLCommons. This organization, which publishes MLPerf benchmarks has published its first round of LLM benchmarking within its MLPerf Training suite as of writing of this paper (June 2023) with MLPerf Inference slated to include LLM in the next version scheduled for release in September 2023. Other organizations have yet to announce their plans for LLM inclusions.

This paper is organized as follows: Section 2 describes difficulties in LLM benchmarking, Sections 3 and 4 give an overview of LLM models and datasets, respectively. Sections 5 and 6 give benchmarking considerations for LLM training and inference, respectively. Section 7 discusses metrics for evaluating LLM performance while Section 8 reviews current state of LLM benchmarking in established benchmarking suites. Finally, Section 9 give a Summary and Conclusions.

2 Difficulties in LLM Benchmarking

Running LLM workloads is challenging because it requires large computational resources. LLM training is usually done on many – tens, hundreds, or even thousands – of GPUs. Therefore, realistic LLM training benchmarking requires an accelerated scale out cluster, which means that an organization needs to put in a substantial investment into its benchmarking compute resources. Beyond raw computational power, a storage system needs to be in place – a commonly used C4 dataset is about 750GB in size – capable of enough throughput for the compute cluster. A common pitfall in LLM performance published so far is measuring throughput only – throughput can often be increased at the cost of worsened convergence and thus does not reflect the actual HW performance. A realistic benchmark needs to take convergence into account, but because the full convergence is too costly it can only be estimated.

LLM inference requires less computational resources, but even then, scale out or at least scale up may still be required owing to the large size of LLM models. These can be pruned and quantized to decrease the size, but that needs to be balanced with accuracy. Beyond the size, realistic LLM inference should generate a reasonable number of words per second – at least a few per second – meaning that just making an

inference benchmark to run is not sufficient to evaluate its deployment performance.

These and other considerations are discussed in more detail in the following sections.

3 LLM Models

Table 1 shows a quick overview of most common LLMs, which are also discussed in more details below. These are built on transformer architecture [3].

Table 1. A selection of noteworthy base LLMs

Model	***Creator***	***Availability***	***Parameters***	***Training Data***	***Architecture***
BERT	Google	Apache 2.0 (commercial)	110M (base), 340M (large)	3.3B words	Bidirectional (not generative) transformer
GPT-3	OpenAI	Proprietary (API only)	175B	300B tokens	Auto-regressive (generative) transformer
LLaMa	Meta	CC BY-NC-SA 4.0 (non-commercial)	7B, 13B, 33B, and 65B	1T tokens (7B, 13B models), 1.4T tokens (33B, 65B models)	Auto-regressive (generative) transformer
LLaMa 2	Meta	Llama 2 community license (commercial)	7B, 13B, 34B, 70B	2T tokens (7B, 13B, 34B, 70B models)	Auto-regressive (generative) transformer
MPT-7B	MosaicML	Apache 2.0 (commercial)	7B	1T tokens	Auto-regressive (generative) transformer
PaLM 2	Google	Proprietary (API only)	Four sizes, count unknown	unknown	Auto-regressive (generative) transformer

BLOOM	BigScience	Big Science RAIL License (commercial)	176B	366B tokens	Auto-regressive (generative) transformer
GPT-J	EleutherAI	Apache 2.0 (commercial)	6B	402B	Auto-regressive (generative) transformer

3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is one of the most widely adopted transformer self-supervised language models. A comparatively simple model to modern standards it has 340M parameters and is not truly a generative model as it is technically an encoder-only bidirectional transformer. However, it serves as the base of so many subsequent models it bears mentioning here. BERT is a masked language model in that it takes the sample dataset and "masks" 15% of all the tokens which it then tries to predict [4]. This was the breakthrough that allowed it to be used on unsupervised datasets such as English Wikipedia (2,500M words) and the BooksCorpus (800M words).

3.2 GPT-3

Generative Pre-trained Transformers (GPT)-3 is a decoder only unidirectional autoregressive model that has 175 billion parameters. This model by OpenAI is the model that powers ChatGPT. This model was trained on 499 billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2) [5]. A previous restriction on many models has been a relatively small context such as only examining a sentence at a time. However, GPT-3 uses 2048-token-long context which is long enough to demonstrate strong zero-shot and few-shot learning on many tasks.

3.3 PaLM

Pathways Language Model (PaLM) is a recent breakthrough from Google that is 540 billion parameters. It gets its name from the novel method in which it is trained, the Pathways system, which enables efficient multi-node training of the model. PaLM has a modified encoder-decoder transformer architecture. It has achieved state-of-the-art results on 28 out of the 29 most widely used English Natural Language Processing (NLP) tasks that "span question-answering tasks (open-domain closed-book variant), cloze and sentence-completion tasks, Winograd-style tasks, in-context reading comprehension tasks, common-sense reasoning tasks, SuperGLUE tasks, and natural language inference tasks" [6]. In addition to this, PaLM is notably achieved state-of-the-art performance on arithmetic and commonsense reasoning task which has traditionally been especially difficult for LLMs.

3.4 LLaMA

Large Language Model Meta AI (LLaMA) was released by Meta AI in February of 2023. This model has a variety of model sizes from 7 billion to 65 billion parameters, with the largest models rivaling PaLM [7]. It draws from an enormous training dataset including webpages (CommonCrawl), open-source repositories (GitHub), Wikipedia in 20 languages, books (Project Gutenberg), scientific papers (ArXiv), and questions and answers (Stack Exchange) [6]. LLaMA 2 was released by Meta AI in July of 2023, LLaMA 2 has a variety of model sizes from 7 billion to 70 billion parameters. LLaMA 2 used public data source, but it didn't specify the source name [30]. Architecturally different from GPT-3, LLaMA/LLaMA 2 uses a SwiGLU activation function, rotary positional embeddings, and root-mean-squared layer normalization.

3.5 GPT-J

Generative Pre-trained Transformer-J (GPT-J) or GPT-J-6B, is a 6 billion parameter model developed by EleutherAI. As the name suggests, it is a GPT-3 like model with a few adjustments. The three main differences are that it uses dense attention, Rotary Position Embeddings, and that the attention and feedforward network were computed in parallel during training [8]. The model is trained on a published dataset that is 800GB consisting of 22 smaller, high-quality datasets [9].

3.6 BLOOM

BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) was created by over 1000 AI researchers to provide an open-source LLM. Unlike a lot of the other LLMs that are only trained on English, BLOOM was trained on the ROOTS corpus which comprises of hundreds of datasets in 46 different natural languages and 13 programming languages [10]. The architecture is modified from Megatron-LM GPT2 and is a decoder-only architecture with ALiBi positional encodings, GeLU activation functions, and layer normalization applied to word embeddings layer.

4 LLM Datasets

Preparing datasets for pre-training the LLMs is a multi-step process. Each of the LLMs described in this paper were pre-trained using datasets prepared uniquely for this purpose. That said, there are also significant commonalities. In this section, we describe these similarities and differences and distill key considerations from a benchmarking perspective.

In general, LLMs are trained in two phases: pre-training and fine-tuning. The pre-training is typically unsupervised and uses a large dataset. The fine-tuning initially was supervised, using a labeled dataset for specific target tasks. This step has now been replaced by few-shot training, using a small number of example prompts. A special case is zero-shot where no examples are used, and the model is directly used with new data. The change from target-specific fine-tuning to few-shot training has been possible by

progressively increasing the scale of the LLM by a factor of ten to a thousand and more. The datasets used for fine-tuning are now largely part of the evaluation and benchmarking stage and span a number of different target tasks.

We focus on the pre-training datasets and describe their evolution over the past several years. The initial GPT [11] used BookCorpus [12]. GPT-2 [13] used WebText created from the Common Crawl web scrape. GPT-3 [14] started with CommonCrawl [15], performed deduplication and added WebText, Books1, Books2, and Wikipedia. GPT-4 [16] has used data from public sources as well as data licensed from third-party sources.

A team from Meta has recently released LLaMA [6]. C4 [15] dataset was used, along with CommonCrawl. The key difference in its training dataset is that it was created from only publicly available data sources. This contrasts with OpenAI's GPT models that use proprietary data licensed from 3rd parties along with publicly available data.

A number of models and chatbots have been released by fine-tuning the LLaMA model: Alpaca, Vicuna, and Koala. Most recently, MosaicML team released the MPT-7B model [17] trained on dataset curated by combining data from various public sources.

Table 2 shows a comparison of data sources used by popular LLMs.

Table 2. Training datasets used by various models

Model	*Training tokens*	*Main data sources for pre-training*	*License*
GPT	0.04T	BookCorpus	Non-commercial
GPT-2	0.4T	WebText, Common Crawl	Proprietary
GPT-3	0.3T	Common Crawl, Books1, Books2, Wikipedia	Proprietary
LLaMA	1T	C4, Common Crawl	Non-commercial (CC BY-NC-SA 4.0)
MPT-7B	1T	Text and code	Commercial (Apache 2.0)

The question of license needs to also be addressed. Most of the LLMs are released under a non-commercial, research use only license. One exception is MosaicML's MPT-7B, which is released under a commercial license.

One key lesson from these modifications to datasets is that limiting the dataset to public-domain accessible sources only does not impact the performance as long as the size of the dataset is large enough. This is good news from a benchmarking perspective. A curated dataset based on public sources can be the basis for use in a benchmarking suite.

5 LLM Training: Benchmarking Considerations

LLM training requires very large computational resources and takes a very long time. As a result, it is very costly. For example, Meta has reported that their LLaMA 65B model training took 21 days on 2,048 NVIDIA A100 GPUs [18]. This is about 1 million GPU hours, which would cost about $2.4 million on AWS [19]. Similarly,

Hugging Face has reported that training their Bloom LLM took more than two-and-a-half months using about 500 GPUs. OpenAI's GPT-3 training cost was estimated at $4 million. This makes realistic benchmarking very difficult because such a high cost cannot be justified for benchmarking work.

Duration of LLM training is determined by the number of tokens processed. For each model, there is an optimal number of tokens needed for training, using more does not improve the results due to overfitting. In general, models with more parameters need more tokens. For example, the GPT-J 6B model needs about 134 billion tokens, while it is about 3.5 trillion tokens for GPT-3 175B. In fact, some of the largest LLMs are limited by insufficient number of quality data to train them.

Optimum number of tokens is determined by testing the model against a chosen accuracy metric. In practice, multiple checkpoints are saved during training, which are later evaluated for performance and best performing ones are chosen for further fine-tuning. This training process is very different from training of convolutional neural nets (CNNs) used for computer vision. There, training iterates multiple times – expressed in number of epochs - over the entire dataset until the loss function is minimized. In LLMs, repeated iterations cause overfitting and degrade performance and thus the training happens within a single epoch on a subset of the dataset.

In general, training time depends on three factors: (i) Number of tokens, (ii) Number of model parameters, and (iii) Number of GPUs used for training.

The training process described above generates an all-purpose model, such models are called foundation models. In practice, fine-tuning is usually applied before deployment. Fine-tuning adapts the model to achieve better performance in a specific domain.

6 LLM Inference: Benchmarking Considerations

Given the extremely high cost of LLM training, one might expect inference to be less costly and easier to run, but that is not necessarily the case. A popular LLM inference service can have thousands of users and needs to have sufficient throughput to generate output quickly. That implies, at a minimum, multi-accelerator deployment, or even scale-out to multiple servers. Thus, a realistic LLM inference benchmark needs to be scalable.

A necessary input for inference is a pre-trained model. This has been straightforward, but with LLMs' extremely high training cost, availability of high-quality checkpoints for models such as GPT-3 175B has been an issue. Beyond the cost, additional issue is that these checkpoints are seen as a competitive advantage for companies that spent large amount of resources creating them. This creates additional barrier for performance evaluation of cutting edge LLMs.

Another issue is model size. For example, GPT-3 175B checkpoints need about 700 GB for storing parameters and about an equal amount for activations. These need to be stored in memory, which means that, at minimum, 16 80GB GPUs are needed to execute such models. To decrease memory requirements and improve compute throughput, pretrained models are usually pruned and quantized. The former refers to removing layers of neural networks, while the latter means converting weights to a lower numerical precision. Both may result in decreased performance and thus the result needs to be carefully tested to ensure that the performance loss is acceptable. Sometimes,

retraining may be needed to recover lost accuracy, but this is impractical for LLMs. In general, this process is complicated, involves using of multiple software tools, and is not captured in current benchmarking tools. Nevertheless, it is an essential part of AI workflow and benchmark creators need expertise in these techniques to create relevant benchmarks.

In LLM inference, pruning and quantization are critical because they decrease size and computational requirements of the models. For pruning, one notable study found that LLMs can be pruned to at least 50% sparsity in one-shot, without any retraining, at minimal loss of accuracy [20]. Similarly, powerful quantizing techniques have been developed, with one of them delivering up to 1.56x speedup and 2x memory reduction [21]. These techniques are critical, because they enable running inference on CPUs and ASIC chips without the need for powerful GPUs.

7 LLM Performance Evaluation

Evaluating the performance of LLMs is an active area of research and discussion. The variety and number of possible evaluations is quite large because of the numerous domains in which LLMs are being used. The range of target tasks include language understanding, question answering systems, text summarization, machine translation and knowledge testing across a variety of disciplines and subjects. In this section, we summarize key metrics used in evaluating the performance of LLMs across the breadth of tasks. Table 3 defines key terms relevant to the evaluation of LLMs.

The straightforward evaluation is done by humans examining the output of the LLMs. These are subjective evaluation of quality of the output and can include factors such as coherence, correctness and contextual relevance. Such evaluations can be averaged by employing many human evaluators, making it a time-consuming and expensive exercise. These considerations have resulted in a number of different metrics specific to target tasks that can be programmatically evaluated.

The intrinsic quality of a language model is measured by its perplexity, the normalized inverse probability of the test set. As an inverse, a lower value of perplexity is better because it indicates a higher corresponding probability value.

For text summarization tasks, the metric used is ROUGE (Recall-Oriented Understudy for Gisting Evaluation). It is a measure of similarity between human-generated, so-called golden annotated summary and the model-generated summary. Evaluation of ROUGE metrics includes 1-gram (ROUGE-1), bi-gram (ROUGE-2) and L-gram (ROUGE-L) based F1 scores, calculated as a harmonic mean of the respective precision and recall values.

Evaluation of machine translation is done using the BLEU (Bilingual Evaluation Understudy) and measures the similarity between a candidate translation generated by the model and a reference translation. BLEU score values are in the range from 0 to 1, and higher values indicate better performance.

LLMs are also measured for knowledge tasks and a common benchmark is the MMLU (Massive Multitask Language Understanding) [22], which includes 57 tasks ranging from elementary mathematics, US history, computer science, law and more.

There are also considerations for measuring diversity in the sense of uniqueness and variety of model output, bias along various dimensions, toxicity, and accuracy in terms of factual correctness. These are all difficult metrics but will play an important role in

increasing trust in AI systems [23].

In summary, many of the metrics described in this section are commonly used in academic and industry publications of new LLMs. These are very good candidates for inclusion in benchmarks for LLMs.

Table 3. Definitions of metrics for LLM benchmarking

Term	*Definition*
Accuracy	The correctness of objective model output
Bias	The systematic prejudice of model output, including stereotypes of certain groups or performance disparities between groups [24]
Coherence	The collective quality of the entire model output; the combined degree to which each part of the model output, given the rest of the output, contributes to the output's quality [25]
Contextual relevance	The degree to which model output is on topic in response to model input
Cost	The explicit cost (e.g., paying for cloud compute) or implicit cost (e.g., the opportunity cost of using owned computing resources) associated with acting on (i.e., training or inferencing) the model
Diversity	The variety of model outputs from varied inputs
Energy Efficiency	The energy consumption required by acting on the model, typically measured per query or per token for inference and in total energy to reach a certain checkpoint (number of tokens ingested) for training
Sustainability	The ability of a model to operate with minimal environmental impact
Throughput	The speed of generation during inference, typically measured in tokens per second
Toxicity	The tendency of a model to output "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion" [26]
Trust	A concept comprised of several measurable metrics – interpretability, reproducibility, and replicability – as well as qualities of a model's creation – transparent knowledge of data and model provenance [27]

8 Current Benchmarking Efforts

8.1 MLPerf Training v3.0 LLM Benchmark Design

MLPerf Training is the first major benchmarking suite that includes an LLM model as of v3.0 released on June 27th, 2023 [28]. MLPerf's model is based on GPT-3 175B model trained on C4 dataset, which is about 350 GB in size and contains 174B tokens. To address the long runtimes, only a small portion of training is executed. Training

starts from an initial checkpoint that has been trained on 12.5B tokens and continues to train on 1.3B tokens to the quality target of 2.69 log perplexity. Thus, the benchmark captures about 0.4% of the full GPT-3 training. Even that still requires substantial resources with reference implementation requiring a minimum of 64 accelerators mainly due to the model's large memory size. To ensure that model is on its way to convergence an accuracy evaluation is performed using about 5% of the total validation set.

8.2 MLPerf Training v3.0 LLM Benchmark results

MLPerf v3.0 results, which include LLM for the first time, are shown in Table 4. The results show that only a few submitters submitted LLM scores; out of 12 submitters in the Closed division and Available categories, only two submitted results: (i) NVIDIA (on their own and also in collaboration with CoreWeave), (ii) Intel-Habana Labs. This indicates that, as designed, this benchmark exceeds resources that most submitters have for their MLPerf work. Indeed, the smallest submission is on 256 Gaudi2 accelerators. The smallest GPU submission is on 768 GPUs, which is 96 NVIDIA DGX H100 servers. On the top end is a 3,584 GPU submission using 448 NVIDIA DGX servers.

The results confirm that LLM training is highly scalable as shown in Figure 1. On NVIDIA servers, going from 768 to 3584 GPUs, 4.67x more, the training time decreases by 4.17x, about 90% efficiency. Because MLPerf timings include accuracy evaluation, this underestimates the actual scaling efficiency. Intel's Gaudi results show even better scaling, of about 95% but over less accelerators – increasing from 256 to 384.

Table 4. LLM results in MLPerf Training v3.0

Submitter	***CPU Qty.***	***Accelerator Type***	***Accelerator Qty.***	***Benchmark Results (min)***
NVIDIA	128	NVIDIA H100-SXM5-80GB	512	64.264
NVIDIA+ CoreWeave	192	NVIDIA H100-SXM5-80GB	768	45.606
NVIDIA	192	NVIDIA H100-SXM5-80GB	768	44.816
NVIDIA+ CoreWeave	384	NVIDIA H100-SXM5-80GB	1536	23.611
NVIDIA+ CoreWeave	896	NVIDIA H100-SXM5-80GB	3584	10.940
Intel-HabanaLabs	64	Habana Gaudi2	256	442.578
Intel-HabanaLabs	96	Habana Gaudi2	384	311.945

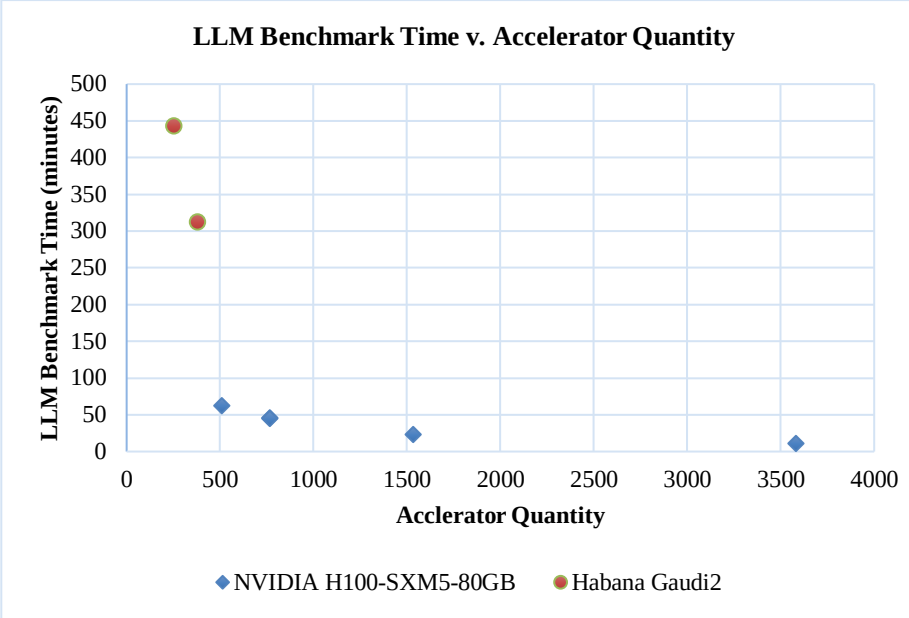

Figure 1. Runtime vs number of accelerators for LLM results in MLPerf Training v3.0

8.3 MLPerf Inference

MLPerf inference has not released an LLM model yet but given that it is usually synchronized with MLPerf training, an LLM inclusion is expected soon. Indeed, its reference code repository contains two LLM implementations, one based on GPT-3 175B corresponding to the MLPerf Training LLM model, while the other is GPT-J 6B, which is not included in the training. This likely reflects the fact that the 175B model is too resource intensive and the 6B model makes it easy to run on smaller computational resources. Because upcoming version of MLPerf Inference is still some time away (September 2023, while this paper is being prepared in June 2023), it is currently unknown whether both models will ultimately be included in the next iteration of the MLPerf Inference suite, but based on publicly available information it looks like a strong possibility.

8.4 Other AI Benchmarking Suites

TPCx-AI is an AI benchmarking suite developed by TPC consortium. Some of the authors of this paper have recently reviewed this suite and compared it to MLPerf [29]. TPCx-AI focuses on the low-end AI, while LLM is on the opposite end of the AI workload spectrum. Current TPCx-AI release does not include LLM nor are there any public indications that it will be included soon.

SpecML is an AI benchmarking effort from another well-established benchmarking consortium, SPEC. Despite being announced some time ago, it has not been released yet and there is no indication that an LLM is being worked on.

9 Summary and Conclusions

This paper has discussed challenges and opportunities in LLM benchmarking as well as reviewed current state of LLM benchmarking in main AI benchmarking suites.

The opportunity part of the equation is quite clear, with exponential growth in LLM model sizes and current rush to deploy these in practice, there is a clear need for representative LLM benchmarks. There are many models to choose from along with many datasets that can be used to train these models. Additionally, there is a wealth of hardware to test on – from high-end GPUs to midsize accelerators, datacenter CPUs all the way down to edge and embedded systems. The amount of reliable LLM benchmarking data is currently very limited meaning that many organizations have to make purchasing and deployment decisions without reliable data.

On the other side, LLM benchmarking also face multiple challenges described in this paper. One of them is the sheer scale of applications meaning that a single model will only address a limited scope of deployments and thus multiple LLM benchmarks are needed to better map out the deployment modes.

Another challenge is a high computational cost of LLM training and inference. The only existing LLM standard benchmark – GPT-3 175B included in MLPerf training – covers only 0.4% of LLM training and yet all the submissions use hundreds of accelerators, which is outside of capabilities of benchmarking setups. The high cost is an inevitable part of LLMs, but using smaller models, such as GPT-J 6B, included in the upcoming MLPerf Inference, can be a way to decrease the computational cost while being a relevant representation of LLM workloads. Ultimately, the practice will show whether the current trend of ever-increasing model sizes will continue or whether practical considerations for deployment will lead to more easily usable model while preserving most of the performance of the cutting edge LLM models.

An important aspect of LLMs is measuring their performance: Our paper discusses some of the most popular options for doing so. While performance testing for, say, image classification, is straightforward, the right metric for generative AI is less obvious. Most likely a combination of several metrics is the best way forward to evaluate usefulness of the results while minimizing bias and other undesirable effects.

In summary, LLM is an extremely active field and benchmarking is currently lagging behind. However, there are efforts under way to improve the current state indicating that the situation is changing. This work surveyed current state of the field of LLM benchmarking pointing out the unique challenges and opportunities for this workload.

References

1. Reuters, accessed June 29, 2023, https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/
2. Google, accessed June 29, 2023, https://blog.google/technology/ai/bard-google-ai-search-updates/
3. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. Adv. Neural Inf. Process. Syst. 2017, 30, 6000–6010.

4. J. Devlin, M. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805.
5. T. Brown, B. Mann, N. Ryder et al, Language Models are Few-Shot Learners, arXiv:2005.14165.
6. S. Narang, A. Chowdhery, Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance, https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html
7. H. Touvron et al, LLaMA: Open and efficient foundation language models, arXiv: 2302.13971v1, 2023.
8. Cerebras, accessed June 29, 2023, https://www.cerebras.net/blog/cerebras-makes-it-easy-to-harness-the-predictive-power-of-gpt-j
9. L. Gao, S. Biderman, S. Black, et al., The Pile: An 800GB Dataset of Diverse Text for Language Modeling, arXiv:2101.00027.
10. T. Le Scao, et al., BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, arXiv:2211.05100.
11. A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, Improving language understanding by generative pre-training, 2018.
12. Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27, 2015.
13. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.
14. T. B. Brown et al, Langage models are few-shot learners, NeurIPS 2020.
15. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research, 21(1):5485–5551, 2020.
16. OpenAI, GPT-4 Technical Report, arXiv:2303.08774v3 2023.
17. MosaicML, Introducing MPT-7B: A new standard for open-source, commercially usable LLMs, May 2023, accessed June 29, 2023, https://www.mosaicml.com/blog/mpt-7b
18. Pursuing groundbreaking scale and accelerating research using Meta's Research SuperCluster, accessed June 29, 2023, https://ai.facebook.com/blog/supercomputer-meta-research-supercluster-2023/
19. ChatGPT and generative AI are booming, but the costs can be extraordinary, accessed June 29, 2023, https://www.cnbc.com/2023/03/13/chatgpt-and-generative-ai-are-booming-but-at-a-very-expensive-price.html#:~:text=Analysts%20and%20technologists%20estimate%20that,could%20cost%20over%20%244%20million
20. E. Frantar and D. Alistarh, SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot, arXiv:2301.00774, 2023.
21. G. Xiao, J. Lin. M. Seznec, H. Wu, J. Demouth, and S. Han, SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models, arXiv:2211.10438, 2023.
22. D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song and H. Steinhardt, Measuring massive multitask language understanding, ICLR, 2021.
23. A. Dholakia, D. Ellison, M. Hodak and D. Dutta, Benchmarking Considerations for Trustworthy and Responsible AI (Panel). In: Nambiar, R., Poess, M. (eds) Performance Evaluation and Benchmarking. TPCTC 2022. Lecture Notes in

Computer Science, vol 13860. Springer, Cham. https://doi.org/10.1007/978-3-031-29576-8_8
24. Bias and Toxicity in Large Language Models, accessed July 18, 2023, https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec14.pdf
25. Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment, arXiv:2303.16634.
26. Perspective, accessed July 18, 2023, https://support.perspectiveapi.com/s/about-the-api-faqs?language=en_US
27. Dholakia, A., Ellison, D., Hodak, M., Dutta, D. (2023). Benchmarking Considerations for Trustworthy and Responsible AI (Panel). In: Nambiar, R., Poess, M. (eds) Performance Evaluation and Benchmarking. TPCTC 2022. Lecture Notes in Computer Science, vol 13860. Springer, Cham. https://doi.org/10.1007/978-3-031-29576-8_8
28. MLCommons, accessed June 29, 2023, https://mlcommons.org/en/training-normal-30/
29. Y. Liu Olesiuk, M. Hodak, D. Ellison, and A. Dholakia, More the Merrier: Comparative Evaluation of TPCx-AI and MLPerf Benchmarks for AI. In: Nambiar, R., Poess, M. (eds) Performance Evaluation and Benchmarking. TPCTC 2022. Lecture Notes in Computer Science, vol 13860. Springer, Cham. https://doi.org/10.1007/978-3-031-29576-8_5
30. H. Touvron et al, Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv: 2307.09288, 2023.