Moxin-7B Technical Report

Pu Zhao[1], **Xuan Shen**[1], **Zhenglun Kong**[2], **Yixin Shen**[3], **Sung-En Chang**[1], **Timothy Rupprecht**[1], **Lei Lu**[1], **Enfu Nan**[1], **Changdi Yang**[1], **Yumei He**[4], **Xingchen Xu**[5], **Yu Huang**[6], **Wei Wang**[6], **Yue Chen**[6], **Yong He**[6], **Yanzhi Wang**[1,7]

[1]Northeastern University, [2]Harvard University,
[3]Cornell University, [4]Tulane University,
[5]University of Washington, [6]Futurewei Technologies, [7]AIBAO LLC

Abstract

Recently, Large Language Models (LLMs) have undergone a significant transformation, marked by a rapid rise in both their popularity and capabilities. Leading this evolution are proprietary LLMs like GPT-4 and GPT-o1, which have captured widespread attention in the AI community due to their remarkable performance and versatility. Simultaneously, open-source LLMs, such as LLaMA and Mistral, have made great contributions to the ever-increasing popularity of LLMs due to the ease to customize and deploy the models across diverse applications. Although open-source LLMs present unprecedented opportunities for innovation and research, the commercialization of LLMs has raised concerns about transparency, reproducibility, and safety. Many open-source LLMs fail to meet fundamental transparency requirements by withholding essential components like training code and data, and some use restrictive licenses whilst claiming to be "open-source," which may hinder further innovations on LLMs. To mitigate this issue, we introduce Moxin 7B, a fully open-source LLM developed in accordance with the Model Openness Framework (MOF), a ranked classification system that evaluates AI models based on model completeness and openness, adhering to principles of open science, open source, open data, and open access. Our model achieves the highest MOF classification level of "open science" through the comprehensive release of pre-training code and configurations, training and fine-tuning datasets, and intermediate and final checkpoints. Experiments show that our model achieves superior performance in zero-shot evaluation compared with popular 7B models and performs competitively in few-shot evaluation.

Homepage: *https://github.com/moxin-org/Moxin-LLM*
Base model: *https://huggingface.co/moxin-org/moxin-7b*
Chat model: *https://huggingface.co/moxin-org/moxin-chat-7b*

1 Introduction

The field of natural language processing has witnessed the most exciting discoveries of the last ten years with the emergence of large language models (LLMs). At the forefront of this evolution are LLMs such as GPT-4 [1], Claude [2], and Gemini [3], which have captured the attention of the AI community due to their performance and versatility. Meanwhile, the recent emergence of openly accessible yet highly capable LLMs such as LLaMA [4], Falcon [5], and Mistral [6] allow researchers and practitioners to easily obtain, customize, and deploy LLMs in more various environments and for more diverse use cases. The trends have made people eagerly asking about what's next and some suggest "a general intelligence" is right around the corner.

Despite the growing influence and accessibility of open-source LLMs, a notable challenge emerged: many model producers restrict visibility and access to their training, fine-tuning, and evaluation processes, including crucial components such as their training code and data [7]. Some model producers even use restrictive licenses whilst claiming to be "open-source." This practice creates barriers for the broader AI research community to study, replicate, and innovate upon advanced LLMs. In parallel, it prevents businesses from fully leveraging open-source models for innovative industrial applications, as its commercialization has raised concerns about transparency, reproducibility, and safety.

To unlock the full potential of LLMs and open innovation, we must return to democratize this research by putting the model into the hands of more researchers and making the datasets the models train on fully open-source. This requires moving beyond the simple sharing of model weights to embrace complete transparency in training, datasets, and implementation detail, which is crucial for fostering a more inclusive and collaborative research environment that can sustain a healthy open-source ecosystem [8].

To achieve this goal, we introduce Moxin 7B, a fully open-source LLM developed by complying with the Model Openness Framework (MOF) introduced by [9]. The MOF provides a systematic ranking classification system to rate AI models based on their completeness and openness, incorporating the principles of open science, open source, open data, and open access. By promoting transparency and reproducibility, the MOF serves as a crucial tool to combat "openwashing" practices and establishes completeness and openness as primary criteria alongside the core tenets of responsible AI. Wide adoption of the MOF will cultivate a more open AI ecosystem, benefiting research, innovation, and adoption of state-of-the-art models.

Our open-source LLM has released pre-training code and configurations, training and fine-tuning data, and intermediate and final checkpoints, aiming to make continuous commitments to fully open-source LLMs. Our model achieves the highest MOF classification level of "open science." It is noteworthy that this commitment to openness has not compromised performance: our base model achieves superior performance in zero-shot evaluation compared with popular 7B models and performs competitively in few-shot evaluation. Remarkably, our chat model can outperform 7B baselines like Llama2-7B-chat. Our Github link is *https://github.com/OminiX-ai/OminiX-LLM*.

2 Related Work

2.1 Models, Tokenizers, and Training

Models. State-of-the-art large language models (LLMs) typically comprise a substantial number of parameters, often approaching or exceeding 100 billion [1], [3], [4]. To facilitate broader accessibility, smaller models with fewer than 20 billion parameters, and even those around 7 billion parameters, have been developed [4], [6], [10]–[13]. In addition, efficiency-enhancing techniques, such as implementing MAMBA-based architectures in Jamba, have been employed to optimize performance [12], [13].

Tokenizers. Tokenizers are essential to convert raw data into a suitable format for model processing. Many contemporary models employ Byte-Pair Encoding (BPE)[14], with OpenAI's `tiktoken` tokenizer[15] being a notable implementation. However, for languages that handle tokens differently from Romance languages, alternatives such as SentencePiece [16] are utilized, as seen in XLNet [17]. Hugging Face offers an excellent summary of state-of-the-art tokenizers with practical examples [18]. Moreover, tokenization extends beyond text modalities; many foundational models now include multimodal capabilities, processing documents, audio, images, and even videos [19]–[22].

Training. To enhance the performance of smaller models beyond their inherent limitations, various training strategies can be employed. A notable example is the application of Mixture of Experts (MoE) training, which has achieved significant success in models like Mixtral [23].

2.2 Data curation methods

Researchers commonly collect large datasets for training language models (LMs)[24] by performing web crawls. However, these datasets often contain undesirable content, necessitating data curation to improve their quality. To enhance model performance[24]–[27], several data curation techniques are widely employed. These include filtering by language [28]–[30], heuristic-based filtering [25], [31], [32], quality filtering [33]–[35], data deduplication [36], [37], and data mixing [38]–[40].

2.3 Open-source datasets

As the scale of LMs has increased in recent years [1], [4], [41], [42], the community has correspondingly curated larger datasets to support their training. Early datasets include the C4 dataset, containing 160 billion tokens, and The Pile [32], which comprises 300 billion tokens. More recently, even larger datasets have been introduced: RefinedWeb [25] with 600 billion tokens, Dolma [43] with 3 trillion tokens, FineWeb [44] with 15 trillion tokens, and RedPajama-v2 [45] containing 30 trillion tokens. In addition to these general-purpose datasets, large domain-specific datasets have also been developed. For instance, StackV2 [46], a code-focused dataset, includes 900 billion tokens, and FineWeb-Edu [44], a high-quality filtered educational text dataset, contains 1.3 trillion tokens.

3 Model Training

3.1 Model Architecture

We opt to extend the Mistral model architecture [6] due to its ability to achieve high performance while maintaining efficient inference speeds. The original Mistral 7B model demonstrates superior performance compared to multiple 7B language models and even outperforms larger models on various evaluation benchmarks. Notably, it surpasses the LLaMA 34B model [47] in tasks such as mathematics and code generation.

The original Mistral model leverages grouped-query attention (GQA)[48] and sliding window attention (SWA)[49]. GQA reduces memory requirements during decoding, allowing for larger batch sizes and higher throughput, and it significantly accelerates inference speed—an essential factor in real-time applications. Meanwhile, SWA effectively handles long sequences without incurring substantial computational overhead. By incorporating these techniques, the model achieves significant improvements in performance and efficiency, which we have adopted in our extended model.

Table 1: Parameter setting.

Parameter	Value
n_layers	36
dim	4096
head_dim	128
hidden_dim	14336
n_heads	32
n_kv_heads	8

Building upon the original Mistral model, which consists of 32 blocks, we have extended the architecture to 36 blocks. Furthermore, we also employ GQA to partition the query heads into multiple groups, each sharing a single key head and value head. This approach interpolates between multi-query attention (MQA) and multi-head attention (MHA) in large language models, striking a balance between the computational speed of MQA and the representational quality of MHA, thereby providing a favorable trade-off. Additionally, our model incorporates a rolling buffer cache with a fixed attention span, effectively limiting cache size and preventing excessive memory usage when processing long sequences.

3.2 Training Data

Data are fundamental to the pre-training of LLMs. Preparing such training data requires careful consideration of multiple challenges, including handling sensitive information, ensuring comprehensive knowledge coverage, and achieving higher efficiency with improved data quality.

In this section, we detail the processes of preparing textual data from general domains and coding data related to programming languages.

3.2.1 Text Data

We use a mix of data from SlimPajama [50] and DCLM-BASELINE [38] as our text training data.

During the training of LLaMA, it was demonstrated that the performance of a 7B model continues to improve even after being trained on more than 1T tokens [51]. Given the outstanding performance of LLaMA, its data collection methodology was rapidly replicated, leading to the release of RedPajama, an open-source dataset containing 1.2 trillion tokens [52].

However, subsequent analyses reveal a significant limitation: some corpora within RedPajama contain a large percentage of duplicate content. The deduplication guidelines in RedPajama operate only within individual data sources, leaving inter-source duplicates largely unaddressed. To improve data quality and training efficiency, SlimPajama was developed as a refined iteration of RedPajama, offering a cleaned and extensively deduplicated version [50].

SlimPajama implements a rigorous two-stage preprocessing pipeline to enhance data quality. In the first stage, short and low-quality documents are removed from RedPajama. Specifically, documents that have fewer than 200 characters after removing punctuation, space symbols, newlines, and tabs are filtered out, as these documents typically contain only metadata and lack useful information. As a result of this step, 1.86% of RedPajama documents are eliminated.

The second step involves removing duplicate data, as deduplication enhances training efficiency and reduces memorization, thereby decreasing the likelihood of generating text solely by recalling training data [25], [36], [53]–[55]. To perform deduplication, document signatures are created using pre-processed, lower-cased 13-grams. Subsequently, MinHashLSH [56] is employed to identify and eliminate duplicates based on a Jaccard similarity threshold of 0.8. Deduplication is performed both within and across data sources. Overall, by pruning 49.6% of the bytes from the RedPajama dataset, the 627B-token SlimPajama dataset is obtained.

Additionally, we utilize the DCLM-BASELINE dataset [38], which is derived from CommonCrawl, a web-crawled dataset [57]. The construction of DCLM-BASELINE involves several steps. First, resiliparse is employed to extract text from CommonCrawl. Second, deduplication is performed using MinHash [58] within a suffix array pipeline [36], [59] and near-duplicate Bloom filtering, which enhances the exact document and paragraph deduplication scheme [43]. Third, recent studies [43], [60], [61] demonstrate that utilizing learnable models as quality filters leads to downstream performance improvements. Consequently, DCLM-BASELINE applies a fastText OH-2.5 combined with an ELI5 classifier score to retain the top 10% of documents.

3.2.2 Coding Data

Programming is crucial for LLMs to support various downstream tasks, such as code completion from natural language descriptions, documentation generation for individual functions, and auto-completion of code snippets. Furthermore, as code is generally better structured and organized than natural language, training on code data may improve the LLM reasoning capabilities [62]. Therefore, We use part of the-stack-dedup dataset [63] during the pretraining.

The Stack comprises more than 6TB of permissively-licensed source code files across 358 programming languages [63]. This carefully curated resource was designed to facilitate the responsible LLMs capable of code generation. It serves for code-generating AI systems to enable the synthesis of programs from natural language descriptions as well as other from code snippets.

To construct the Stack dataset, 220.92 million active GitHub repositories were collected from event archives published between 2015 and 2022 on GHArchive. Of these repositories, only 137.36 million were publicly accessible on GitHub, resulting in 51.76 billion downloaded files. After initial filtering, 5.28 billion unique files were identified, with an uncompressed size of 92.36 TB.

To ensure data quality, near-deduplication was implemented within the preprocessing pipeline in addition to exact deduplication. Specifically, MinHash with 256 permutations was computed for all documents, and Locality Sensitive Hashing was employed to identify clusters of duplicates. Within these clusters, Jaccard similarities were calculated to detect near-duplicates using a similarity threshold of 0.85. Approximately 40% of permissively licensed files were identified as (near-)duplicates and subsequently removed.

3.2.3 Capability Enhancement

LLMs are expected to demonstrate capabilities such as reasoning, mathematical problem-solving, and knowledge memorizing. However, a significant challenge lies in that, in the pre-training process, high-quality capability-related data is sparsely distributed in the entire corpus, and thereby it is difficult for models to be proficient at these above-mentioned capabilities. Previous research, such as work on Qwen [10], GLM-130B [64], Nemotron-4 [65], has tried to incorporate instruction-based or high-quality data during the pre-training stage to enhance these abilities. In our study, we collect open-source data from HuggingFace, primarily utilizing the training datasets of various evaluation benchmarks such as MMLU [66] and HellaSwag [67]. These data are used experimentally to investigate the relationship between high-quality, capability-focused training data and model performance.

3.3 Training Configuration

The total number of tokens used for pre-training our Moxin-7B model is over 1.5T, and the pre-training process consists of three phases. In the first phase, we use pre-training corpora with the context length of 2k. In the second phase, we use pre-training corpora with the context length of 4k. In the third phase, we utilize the capability-specific enhancement data. We provide the model performance with only the first two phases and also with all three phases to validate the performance of the third phase.

We use Colossal-AI [68] as our training framework. Colossal-AI is a unified deep learning system that provides the fullest set of acceleration techniques for the AI community. With its modular design, ColossalAI allows for a free combination of these techniques to achieve the best training speedup. Colossal-AI's optimized parallelism and heterogeneous training methods are employed to achieve superior system performance compared to baseline systems. These methods are provided through user-friendly APIs, requiring minimal code modifications.

During training, AdamW [69] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1e^{-8}$ and weight decay = 0.1 is used to optimize the model. We use the cosine learning rate decay and the learning rate decays to 10% of its maximum. Learning Rate is set to $2e^{-6}$.

3.4 Alignment

Following the pre-training phase, we fine-tune the model into a helpful and harmless AI assistant. In our Alignment stage, we mainly use supervised fine-tuning (SFT), during which we fine-tune the model to follow diverse human instructions by high-quality instruction data. We use the Tulu v2 dataset [70] for instruction tuning. The dataset consists of a mix of FLAN, Open Assistant 1, ShareGPT, GPT4-Alpaca, LIMA, and so on.

3.5 Long-Context

To deal with the long-context problem, our model leverages grouped-query attention (GQA) [48], sliding window attention (SWA) [49], and Rolling Buffer Cache [6]. GQA reduces the memory requirement during decoding, allowing for higher batch sizes hence higher throughput.

Besides, SWA can handle longer sequences more effectively at a reduced computational cost, thereby alleviating a common limitation in LLMs. SWA exploits the stacked layers of a transformer to attend information beyond the window size W. At the last layer, with SWA, using a window size of $W = 4096$, we have a theoretical attention span of approximately $14K$ tokens or above.

Our model adopts Rolling Buffer Cache which limits the cache size using a rolling buffer cache with a fixed attention span. The cache has a fixed size of W, and the keys and values for the timestep i are stored in position i mod W of the cache. As a result, when the position i is larger than W, past values in the cache are overwritten, and the size of the cache stops increasing. On a sequence length of $32k$ tokens, this reduces the cache memory usage by $8\times$, without impacting the model quality.

With the above techniques, our model can support $32K$ context length with fast inference and low memory cost.

4 Evaluation

We conducted comprehensive performance comparisons against leading language models of comparable scale, including Mistral-7B [6], LLaMA 2-7B [51], Gemma-7B [41], and Qwen v2-7B [11]. These models were selected based on their demonstrated excellence within the 7B or 8B category and represent diverse development approaches from various research organizations worldwide. To ensure a robust evaluation, we re-run all benchmarks with the same evaluation pipeline for fair comparisons. Specifically, we use lm-evaluation-harness [71] and opencompass [72] for evaluation.

Lm-evaluation-harness provides a unified framework to test generative language models on a large number of different evaluation tasks. It supports over 60 standard academic benchmarks for LLMs, with hundreds of subtasks and variants implemented. This framework is versatile as it extends to models implemented through various architectures, including transformers (including quantization via AutoGPTQ [73]), GPT-NeoX [74], and Megatron-DeepSpeed [75], all unified through a flexible, tokenization-agnostic interface. The framework is reliable, as evidenced by serving as the backend for HuggingFace's popular Open LLM Leaderboard and being utilized by dozens of organizations, including NVIDIA, Cohere, BigScience, BigCode, Nous Research, and Mosaic ML.

To complement, we also employed openCompass. This framework performs an in-depth and holistic assessment of large language models structured around eight fundamental dimensions of language model capabilities: language comprehension, knowledge precision, logical deduction, creative ideation, mathematical problem-solving, programming proficiency, extended text analysis, and intelligent agent engagement.

4.1 Evaluation Tasks

We evaluate the model performance on various tasks below.

- AI2 Reasoning Challenge (ARC) [76] - a set of genuine grade-school level, multiple-choice science questions, assembled to encourage research in advanced question-answering. The dataset is partitioned into a Challenge Set (ARC-C) and an Easy Set (ARC-E), where the former contains only questions answered incorrectly by both a retrieval-based algorithm and a word co-occurrence algorithm.
- HellaSwag [67] - a test of commonsense natural language inference, which is easy for humans (95%) but challenging for SOTA models. It consists of 70,000 multiple-choice questions. Each question presents a scenario followed by four possible outcomes, asking the model to select the most reasonable conclusion.
- MMLU [77] - a test to measure a text model's multitask accuracy. The test covers 57 tasks, including elementary mathematics, US history, computer science, law, etc.
- Winogrande [78] - an adversarial and difficult Winograd benchmark at scale, for commonsense reasoning. It contains 44,000 multiple-choice questions with two options each. It requires the model to choose the appropriate entity word for the pronoun in the descriptive text based on the scenario.
- PIQA [79] - the task of physical commonsense reasoning and a corresponding benchmark dataset Physical Interaction: Question Answering (PIQA). Physical commonsense knowledge is a major challenge on the road to true AI-completeness, including robots that interact with the world and understand natural language. PIQA focuses on everyday situations with a preference for atypical solutions.

4.2 Evaluation Results

We name the initial model as Moxin-7B-original, which presents the foundation model before fine-tuning on the training data of the evaluation datasets. After subsequent partial fine-tuning of Moxin-7B-original on the training data of the evaluation datasets, we developed Moxin-7B-finetuned, enabling direct assessment of how targeted fine-tuning affects model performance.

4.2.1 Zero-Shot Evaluation

We report the result of base models for zero-shot evaluation in Table 2. The tasks are listed below. After training with the training data of evaluation tasks, our Moxin-7B-finetuned can achieve superior performance compared with state-of-the-art (SOTA) baselines. This significant increase from the base model demonstrates the effectiveness of our fine-tuning approach. The improved performance is particularly notable on complex reasoning tasks like PIQA, where the score increased from 78.07% to 82.24%, matching or exceeding several leading models. Consequently, our models emerge as an excellent candidate for real-world applications.

- AI2 Reasoning Challenge (0-shot)

- AI2 Reasoning Easy (0-shot)
- HellaSwag (0-shot)
- PIQA (0-shot)
- Winogrande (0-shot)

Table 2: Performance comparison for various models in zero-shot evaluation.

Models	HellaSwag	WinoGrade	PIQA	ARC-E	ARC-C	Ave
Mistral - 7B	80.39	73.4	82.15	78.28	52.22	73.29
LLaMA 2 - 7B	75.99	69.06	79.11	74.54	46.42	69.02
LLaMA 2 - 13B	79.37	72.22	80.52	77.4	49.06	71.71
LLaMA 3.1 - 8B	78.92	74.19	81.12	81.06	53.67	73.79
gemma - 7b	80.45	73.72	80.9	79.97	54.1	73.83
Qwen v2 - 7B	78.9	72.38	79.98	74.71	50.09	71.21
internlm2.5 - 7b	79.14	77.9	80.52	76.16	51.37	73.02
Baichuan2 - 7B	72.25	67.17	77.26	72.98	42.15	66.36
Yi-1.5-9B	77.86	73.01	80.74	79.04	55.03	73.14
deepseek - 7B	76.13	69.77	79.76	71.04	44.8	68.3
Moxin - 7B - original	72.06	66.31	78.07	71.47	48.15	67.21
Moxin - 7B - finetune	80.03	75.17	82.24	81.12	58.64	75.44

4.2.2 Few-Shot Evaluation

Table 3 presents our zero-shot evaluation results across multiple benchmark tasks. The tasks and their few-show settings are listed below. Thanks to its rigorous and high-quality training corpus, our model demonstrates a remarkable competitive edge in tasks that involve language understanding and knowledge application. Our Moxin-7B-original achieves superior performance than LLaMA2-7B in this scenario. After training with the training data of evaluation tasks, our Moxin-7B-finetuned can achieve competitive performance compared with SOTA baselines.

Consequently, our models emerge as an excellent choice for a multitude of real-world applications where the reliance on robust language comprehension and extensive knowledge is paramount.

- AI2 Reasoning Challenge (25-shot)
- HellaSwag (10-shot)
- MMLU (5-shot)
- Winogrande (5-shot)

Table 3: Performance comparison for various models in few-shot evaluation.

model	ARC-C	hellaswag	mmlu	WinoGrade	Ave
Mistral - 7B	57.59	83.25	62.42	78.77	70.51
LLaMA 3.1 - 8B	54.61	81.95	65.16	77.35	69.77
LLaMA 3 - 8B	55.46	82.09	65.29	77.82	70.17
LLaMA 2 - 7B	49.74	78.94	45.89	74.27	62.21
Qwen 2 - 7B	57.68	80.76	70.42	77.43	71.57
gemma - 7B	56.48	82.31	63.02	78.3	70.03
internlm2.5 - 7B	54.78	79.7	68.17	80.9	70.89
Baichuan2 - 7B	47.87	73.89	54.13	70.8	61.67
Yi-1.5-9B	58.36	80.36	69.54	77.53	71.48
Moxin - 7B - original	53.75	75.46	59.43	70.32	64.74
Moxin - 7B - finetuned	59.47	83.08	60.97	78.69	70.55

4.3 Alignment Evaluation

We evaluate the alignment performance on MTBench [80]. It is a two-round conversation dataset with 80 questions. It covers eight dimensions (reasoning, roleplay, math, coding, writing, humanities, STEM, and information extraction) with 10 questions for each dimension. The model needs to answer the first question and then refine its previous response following additional specific instructions. We use GPT-4 as a judge model to provide scores (between 1-10) for the quality of responses. Our Moxin-7B-chat achieves superior performance on MTbench compared with baselines, as shown in Table 4.

Table 4: Performance for various chat models.

Model	MTbench
Moxin-7B-chat	**6.42**
Llama 2 13B Chat	6.65
Vicuna 13B	6.57
Llama 2 7B Chat	6.27
Vicuna 7B	6.17
Alpaca 13B	4.53

5 Conclusion

The field of Large Language Models has witnessed a significant shift toward open-source development, fostering innovation within the AI community. However, a critical challenge emerges: many purportedly open-source models withhold essential components necessary for full understanding and reproducibility, creating barriers that limit both academic advancement and commercial adoption. This does not not only hamper scientific progress but also prevent businesses from fully leveraging these models for innovative applications, ultimately diminishing potential societal benefits and economic value creation. To address these limitations, we introduce Moxin 7B, a fully open-source language model developed in accordance with the Model Openness Framework (MOF), providing comprehensive access to pre-training code, configurations, training and fine-tuning datasets, and all intermediate checkpoints. Our evaluation results demonstrate that the Moxin 7B achieves superior zero-shot evaluation results compared to popular 7B models while maintaining competitive few-shot capabilities. We wish to see more work that establishes new standard for reproducible research in language model development, fostering a more inclusive and economically vibrant AI ecosystem.

References

[1] J. Achiam, S. Adler, S. Agarwal, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] Anthropic, *The claude 3 model family: Opus, sonnet, haiku*, https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[3] G. Team, R. Anil, S. Borgeaud, *et al.*, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[4] A. Dubey, A. Jauhri, A. Pandey, *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[5] T. Prest, P.-A. Fouque, J. Hoffstein, *et al.*, "Falcon," *Post-Quantum Cryptography Project of NIST*, 2020.

[6] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[7] R. Bommasani, K. Klyman, S. Longpre, *et al.*, "The foundation model transparency index," *arXiv preprint arXiv:2310.12941*, 2023.

[8] S. Kapoor, R. Bommasani, K. Klyman, *et al.*, "On the societal impact of open foundation models," *arXiv preprint arXiv:2403.07918*, 2024.

[9] M. White, I. Haddad, C. Osborne, A. Abdelmonsef, S. Varghese, *et al.*, "The model openness framework: Promoting completeness and openness for reproducibility, transparency and usability in ai," *arXiv preprint arXiv:2403.13784*, 2024.

[10] J. Bai, S. Bai, Y. Chu, *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[11] A. Yang, B. Yang, B. Hui, *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[12] O. Lieber, B. Lenz, H. Bata, *et al.*, "Jamba: A hybrid transformer-mamba language model," *arXiv preprint arXiv:2403.19887*, 2024.

[13] J. Team, B. Lenz, A. Arazi, *et al.*, "Jamba-1.5: Hybrid transformer-mamba models at scale," *arXiv preprint arXiv:2408.12570*, 2024.

[14] R. Sennrich, “Neural machine translation of rare words with subword units,” *arXiv preprint arXiv:1508.07909*, 2015.

[15] O. Team, *Tiktoken*, 2022. [Online]. Available: https://github.com/openai/tiktoken.

[16] T. Kudo, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *arXiv preprint arXiv:1808.06226*, 2018.

[17] Z. Yang, “Xlnet: Generalized autoregressive pretraining for language understanding,” *arXiv preprint arXiv:1906.08237*, 2019.

[18] H. F. Team, “Summary of the tokenizers,” 2024. [Online]. Available: https://github.com/huggingface/transformers/blob/main/docs/source/en/tokenizer_summary.md.

[19] M. Reid, N. Savinov, D. Teplyashin, *et al.*, “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” *arXiv preprint arXiv:2403.05530*, 2024.

[20] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, “Video-chatgpt: Towards detailed video understanding via large vision and language models,” *arXiv preprint arXiv:2306.05424*, 2023.

[21] H. Zhang, X. Li, and L. Bing, “Video-llama: An instruction-tuned audio-visual language model for video understanding,” *arXiv preprint arXiv:2306.02858*, 2023.

[22] D. Zhang, Y. Yu, C. Li, *et al.*, “Mm-llms: Recent advances in multimodal large language models,” *arXiv preprint arXiv:2401.13601*, 2024.

[23] A. Q. Jiang, A. Sablayrolles, A. Roux, *et al.*, “Mixtral of experts,” *arXiv preprint arXiv:2401.04088*, 2024.

[24] B. Mann, N. Ryder, M. Subbiah, *et al.*, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, vol. 1, 2020.

[25] G. Penedo, Q. Malartic, D. Hesslow, *et al.*, “The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only,” *arXiv preprint arXiv:2306.01116*, 2023.

[26] J. W. Rae, S. Borgeaud, T. Cai, *et al.*, “Scaling language models: Methods, analysis & insights from training gopher,” *arXiv preprint arXiv:2112.11446*, 2021.

[27] G. Wenzek, M.-A. Lachaux, A. Conneau, *et al.*, “Ccnet: Extracting high quality monolingual datasets from web crawl data,” *arXiv preprint arXiv:1911.00359*, 2019.

[28] L. Xue, “Mt5: A massively multilingual pre-trained text-to-text transformer,” *arXiv preprint arXiv:2010.11934*, 2020.

[29] C. Raffel, N. Shazeer, A. Roberts, *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[30] A. Conneau and G. Lample, “Cross-lingual language model pretraining,” *Advances in neural information processing systems*, vol. 32, 2019.

[31] M. Chen, J. Tworek, H. Jun, *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.

[32] L. Gao, S. Biderman, S. Black, *et al.*, “The pile: An 800gb dataset of diverse text for language modeling,” *arXiv preprint arXiv:2101.00027*, 2020.

[33] N. Sachdeva, B. Coleman, W.-C. Kang, *et al.*, “How to train data-efficient llms,” *arXiv preprint arXiv:2402.09668*, 2024.

[34] S. Longpre, G. Yauney, E. Reif, *et al.*, “A pretrainer’s guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity,” *arXiv preprint arXiv:2305.13169*, 2023.

[35] N. Du, Y. Huang, A. M. Dai, *et al.*, “Glam: Efficient scaling of language models with mixture-of-experts,” in *International Conference on Machine Learning*, PMLR, 2022, pp. 5547–5569.

[36] K. Lee, D. Ippolito, A. Nystrom, *et al.*, “Deduplicating training data makes language models better,” *arXiv preprint arXiv:2107.06499*, 2021.

[37] A. Agarwal, H. S. Koppula, K. P. Leela, *et al.*, “Url normalization for de-duplication of web pages,” in *Proceedings of the 18th ACM conference on information and knowledge management*, 2009, pp. 1987–1990.

[38] J. Li, A. Fang, G. Smyrnis, *et al.*, “Datacomp-lm: In search of the next generation of training sets for language models,” *arXiv preprint arXiv:2406.11794*, 2024.

[39] A. Albalak, L. Pan, C. Raffel, and W. Y. Wang, “Efficient online data mixing for language model pre-training,” in *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[40] Z. Shen, T. Tao, L. Ma, *et al.*, “Slimpajama-dc: Understanding data combinations for llm training,” *arXiv preprint arXiv:2309.10818*, 2023.

[41] G. Team, T. Mesnard, C. Hardin, *et al.*, “Gemma: Open models based on gemini research and technology,” *arXiv preprint arXiv:2403.08295*, 2024.

[42] A. Chowdhery, S. Narang, J. Devlin, *et al.*, “Palm: Scaling language modeling with pathways,” *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[43] L. Soldaini, R. Kinney, A. Bhagia, *et al.*, “Dolma: An open corpus of three trillion tokens for language model pretraining research,” *arXiv preprint arXiv:2402.00159*, 2024.

[44] G. Penedo, H. Kydlíček, A. Lozhkov, *et al.*, “The fineweb datasets: Decanting the web for the finest text data at scale,” *arXiv preprint arXiv:2406.17557*, 2024.

[45] M. Ostendorff, P. O. Suarez, L. F. Lage, and G. Rehm, “Llm-datasets: An open framework for pretraining datasets of large language models,”

[46] A. Lozhkov, R. Li, L. B. Allal, *et al.*, “Starcoder 2 and the stack v2: The next generation,” *arXiv preprint arXiv:2402.19173*, 2024.

[47] B. Roziere, J. Gehring, F. Gloeckle, *et al.*, “Code llama: Open foundation models for code,” *arXiv preprint arXiv:2308.12950*, 2023.

[48] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, “Gqa: Training generalized multi-query transformer models from multi-head checkpoints,” *arXiv preprint arXiv:2305.13245*, 2023.

[49] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” *arXiv preprint arXiv:2004.05150*, 2020.

[50] D. Soboleva, F. Al-Khateeb, R. Myers, J. R. Steeves, J. Hestness, and N. Dey, *SlimPajama: A 627B token cleaned and deduplicated version of RedPajama*, https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. [Online]. Available: https://huggingface.co/datasets/cerebras/SlimPajama-627B.

[51] H. Touvron, T. Lavril, G. Izacard, *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[52] M. Weber, D. Fu, Q. Anthony, *et al.*, “Redpajama: An open dataset for training large language models,” *arXiv preprint arXiv:2411.12372*, 2024.

[53] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos, “Semdedup: Data-efficient learning at web-scale through semantic deduplication,” *arXiv preprint arXiv:2303.09540*, 2023.

[54] *Large-scale near-deduplication behind bigcode*, 2023. [Online]. Available: https://huggingface.co/blog/dedup.

[55] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” *arXiv preprint arXiv:1904.09751*, 2019.

[56] J. Leskovec, A. Rajaraman, and J. Ullman, *Mining of massive datasets, cambridge university press, cambridge*, 2014.

[57] J. M. Patel and J. M. Patel, “Introduction to common crawl datasets,” *Getting structured data from the internet: running web crawlers/scrapers on a big data production scale*, pp. 277–324, 2020.

[58] A. Z. Broder, “On the resemblance and containment of documents,” in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, IEEE, 1997, pp. 21–29.

[59] *Fineweb*, 2024. [Online]. Available: https://huggingface.co/datasets/HuggingFaceFW/fineweb.

[60] D. Brandfonbrener, H. Zhang, A. Kirsch, J. R. Schwarz, and S. Kakade, “Color-filter: Conditional loss reduction filtering for targeted language model pre-training,” *arXiv preprint arXiv:2406.10670*, 2024.

[61] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. Toshev, and V. Shankar, “Data filtering networks,” *arXiv preprint arXiv:2309.17425*, 2023.

[62] D. Groeneveld, I. Beltagy, P. Walsh, *et al.*, “Olmo: Accelerating the science of language models,” *arXiv preprint arXiv:2402.00838*, 2024.

[63] D. Kocetkov, R. Li, L. B. Allal, *et al.*, “The stack: 3 tb of permissively licensed source code,” *arXiv preprint arXiv:2211.15533*, 2022.

[64] A. Zeng, X. Liu, Z. Du, *et al.*, "GLM-130b: An open bilingual pre-trained model," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=-Aw0rrrPUF.

[65] J. Parmar, S. Prabhumoye, J. Jennings, *et al.*, "Nemotron-4 15b technical report," *arXiv preprint arXiv:2402.16819*, 2024.

[66] D. Hendrycks, C. Burns, S. Basart, *et al.*, *Measuring massive multitask language understanding*, 2021. arXiv: 2009.03300 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2009.03300.

[67] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[68] S. Li, H. Liu, Z. Bian, *et al.*, "Colossal-ai: A unified deep learning system for large-scale parallel training," in *Proceedings of the 52nd International Conference on Parallel Processing*, 2023, pp. 766–775.

[69] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[70] H. Ivison, Y. Wang, V. Pyatkin, *et al.*, "Camels in a changing climate: Enhancing lm adaptation with tulu 2," *arXiv preprint arXiv:2311.10702*, 2023.

[71] L. E. H. Team, *Lm evaluation harness*, Accessed: Summer 2024, 2024. [Online]. Available: https://github.com/EleutherAI/lm-evaluation-harness.

[72] O. C. Team, *Open compass*, Accessed: Summer 2024, 2024. [Online]. Available: https://github.com/open-compass/opencompass.

[73] A. Team, *Autogptq: An user-friendly llms quantization package*, Accessed: Spring 2024, 2024. [Online]. Available: https://github.com/AutoGPTQ/AutoGPTQ.

[74] S. Black, S. Biderman, E. Hallahan, *et al.*, "Gpt-neox-20b: An open-source autoregressive language model," *arXiv preprint arXiv:2204.06745*, 2022.

[75] S. L. Song, B. Kruft, M. Zhang, *et al.*, "Deepspeed4science initiative: Enabling large-scale scientific discovery through sophisticated ai system technologies," *arXiv preprint arXiv:2310.04610*, 2023.

[76] P. Clark, I. Cowhey, O. Etzioni, *et al.*, "Think you have solved question answering? try arc, the ai2 reasoning challenge," *arXiv:1803.05457v1*, 2018.

[77] T. van der Weij, F. Hofstätter, O. Jaffe, S. F. Brown, and F. R. Ward, "Ai sandbagging: Language models can strategically underperform on evaluations," *arXiv preprint arXiv:2406.07358*, 2024.

[78] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *Communications of the ACM*, vol. 64, no. 9, pp. 99–106, 2021.

[79] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, *Piqa: Reasoning about physical commonsense in natural language*, 2019. arXiv: 1911.11641 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1911.11641.

[80] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.