Reading Between the Lines: Commonsense Reasoning in Small Language Models

1st Wasif Feroze
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
wasif.feroze@std.uestc.edu.cn

2nd Hong Qu
School of Computer Science and Engineering
University of Electronic Science and Technology of China;
and Tianfu Jiangxi Laboratory
Chengdu, China
hongqu@uestc.edu.cn

3rd Muhammad Shahid
Department of Computer Sciences
Bahria University
Islamabad, Pakistan
mshahid.bulc@bahria.edu.pk

4th Shaohuan Cheng
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
shaohuancheng@std.uestc.edu.cn

2024 10th International Conference on Computer and Communications (ICCC) | 979-8-3315-0707-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCC62609.2024.10941834

***Abstract*—Recently, large language models have been scaled down from large to smaller parameters. Large language models have generalized to many tasks with pre-training, and these also excelled in commonsense reasoning with targeted fine-tuning. Commonsense reasoning is the capability to make judgments and draw conclusions based on everyday knowledge that humans typically acquire through life experiences. Reasoning ability in language models involves understanding implicit relationships, contextual cues, and causal connections in various scenarios. Despite the progress of large models in many tasks, commonsense reasoning has proved challenging in few-shot settings. In this paper, we propose the evaluation of small language models for commonsense reasoning using the instruction tuning method. We performed experiments on two datasets for commonsense reasoning and evaluated the performance of the models with different quantization processes in one-shot settings. Our results show that the model demonstrates promising results; however, further fine-tuning is required to enhance their commonsense reasoning abilities. Our study contributes to understanding the potential and limitations of small language models.**

***Index Terms*—small language models, commonsense reasoning, instruction tuning, natural language understanding, large language models**

I. Introduction

Commonsense reasoning is the process by which machine intelligence understands implicit reasoning knowledge that may not explicitly present any commonsense clues in the given data. Acquiring commonsense knowledge is challenging for artificial intelligence because it differs from data in large corpora form for other general NLP fields [1]. Therefore, evaluating language models for such knowledge is crucial because of the complexity of commonsense reasoning tasks. In recent years, large language models (LLMs) have significantly advanced natural language understanding (NLU) benchmarks [2]. These benchmarks are vital for assessing language models' ability to understand language text. Given the significance of commonsense reasoning and NLU, our research focuses on machine reading comprehension (MRC) with commonsense reasoning tasks. To achieve this, we evaluated the open Small Language Model (SLM) on datasets that combine these two tasks.

Language models have undergone a significant transformation following the methodology shift from recurrent neural network architectures to transformer architectures [3], enabling extensive scalability. This paradigm shift has led to the spread of larger models in the NLP research community, sparking debate on quantifying the scale of models in terms of parameters. While models with fewer than seven billion parameters are generally considered small, and those exceeding this threshold are termed LLMs, it is worth noting that even one billion parameters would have been considered large by past standards. To explore the capabilities of smaller models, we focus on evaluating the performance of an open SLM in commonsense reasoning and MRC tasks using instruction-tuning-based evaluation. Specifically, we employed Phi-3 [4], an SLM introduced by Microsoft, and implemented prompt tuning evaluation techniques to assess its commonsense reasoning abilities. Instruction-tuning and prompting techniques have proven highly effective in eliciting various behaviors in LLMs [5]. Our research aims to investigate whether these approaches are similarly effective in improving commonsense reasoning capabilities in smaller models. This study provides insights into the ability of SLMs to understand and involve commonsense reasoning in MRC tasks and presents results

This work was supported in part by the National Science Foundation of China under Grant 62306063 and in part by the Young Scientists Fund of the Natural Science Foundation of Sichuan Province under Grant 2024NSFSC1469. The code for this paper is available at https://github.com/wasifferoze/small-lm-csr-evaluation.

that demonstrate the effectiveness of the instruction-following paradigm.

Our framework for prompt tuning-based evaluation of SLMs is straightforward but proved very effective because we can obtain an aligned response from small models for MRC commonsense reasoning, as these models are usually good at following general instructions to present their common knowledge to human agents; however, commonsense reasoning is a complex task and aligning a SLMs to handle questions that require some commonsense reasoning demands some iteration of interaction with the model and derives the best prompt that makes this model follow instructions. After this prompt design, we evaluated these models on our two chosen datasets of the MRC task, which have commonsense reasoning. Our results show that these small models can follow instructions; however, understanding commonsense reasoning using the prompting technique is very challenging, and we strongly argue that these models require fine-tuning on the target datasets to enhance their capability to comprehend commonsense reasoning questions.

Our paper is organized as follows. We briefly introduce the most relevant theories in Section II. Section III presents more information on our prompting-based evaluation and instruction-tuning framework, the architecture of SLMs, and the quantization process of these SLMs. Section IV presents technical details of the methodology, experimental setup, and dataset information. Section V presents the most significant results of our framework achieved with SLM for commonsense reasoning and some critics of performance. Finally, we present the research conclusions, summarize the limitations, and offer future research directions in the Section VI.

II. Related Work

A. Commonsense Reasoning

There are several reasoning types in automated machine intelligence, such as abductive, inductive, commonsense, quantitative, and symbolic reasoning. Our focus in this research is commonsense reasoning, and we only discuss benchmarks and the literature on this reasoning type [1]. Commonsense reasoning also has several types of categories: social commonsense reasoning, often called folk psychology; physical commonsense reasoning, also referred to as naive physics in the more traditional way of reasoning; and temporal commonsense reasoning, which is used to study the reasoning aspect of time and events. There are several benchmarks and datasets for all these types of commonsense reasoning, such as CommonsenseQA [6], a question-answering benchmark for general commonsense reasoning that does not focus on any particular commonsense reasoning category. Other such datasets without a particular focus are COPA [7] and WinoGrande [8]. The famous benchmark datasets that mainly focus on Naive Physics and Folk Psychology are PIQA [9] and SIQA [10], both of which are question-answering tasks. PIQA discusses the physical reasoning aspects of objects. SIQA focuses more on social interactions in humans and offers a unique aspect of reasoning for studying the behavior of human sociology. Many previous studies have focused on commonsense reasoning and have evaluated their models on such datasets. However, little attention has been paid to the distinct relationship between MRC and commonsense reasoning tasks, so this is a large gap in the current state of research, and we present our study to fill this gap by evaluating SLMs on these datasets such as ReCoRD [11], and CosmosQA [12].

B. Small and Large Language Models

In NLP research, unsupervised pre-training techniques [14] with transformer architectures [3] have provided scalability opportunities for language models since the inception of pre-training in the field of computer vision [13]. This fundamental paradigm shift in the NLP community has resulted in an influx of language models, more often with large parameters from GPTs [5], T5 [15], GPT3 [5], and FLAN [16]. The GPT-3 model has been applied in many applications, such as ChatGPT, the most hyped-up tool in the history of AI. This considerable AI adaptation of the general public and hype forced researchers to disrupt the hegemony of GPT-3 models, which resulted in LLaMA [17] models because these models were open-sourced and had open weights. This open-source mindset provided many researchers opportunities to offer innovation by modifying LLaMA-based models, so many models presented open weights with different sizes of parameters, generally with large parameters such as more than 7B according to our set standard for large versus SLMs; however, most small-scale models are presented in this literature. Most of these SLMs are modified versions of the LLaMA models, such as Phi-1 and Phi-2. Other related SLMs are Pythia [18], StableLM [19], and Gemma [20].

C. Instruction Tuning

After introducing the pre-training paradigm and with the ever-increasing scale of pre-trained models, a new parameter-efficient technique was introduced to explore the capability of LLMs [23]. This technique is called prompt tuning and involves instructing the models without updating their parameters using instructions and obtaining the required answers from the models. For this purpose, many prompting techniques are used, such as static, automated, and dynamic prompting. Some instruction-tuning frameworks, such as InstructGPT [25], PromptSource [24], Flan Collection [21], and SuperNatInstruct [22], offer task templates for classification, sentiment analysis, and text generation tasks. We are using a static prompting technique to evaluate SLM, and no other work on our targeted datasets has performed this evaluation for SLMs.

III. Methods

A. Overview

The overview of our methodology is that we preprocessed our target dataset into an instruction-tuning format, where instruct-based models perform excellently in such a setting. The next most crucial component of this framework is SLM, which is used to obtain the output of the input provided as a

TABLE I
PHI-3 ARCHITECTURE'S DETAIL

Parameter Name	Detail
Tokenizer	LLaMA 2
Vocabulary Size	32064
Pre-train Tokens	3.3T
Model Size	3.8B
Hidden Dimensions	3072
No. of Heads	32
No. of Layers	32

prompt to the model. The next component is the quantization requirement checker, which puts restrictions on whether to utilize a quantized SLM or the model's standard weights with default precision type. Finally, we captured the models' responses for further processing, such as calculating the evaluation scores and quantifying our results. The overall architecture of the prompting framework is illustrated in Fig. 1. Further, we present the information about our SLM and quantization process in the following subsections.

B. Small Language Model

For our language model, we used Phi-3 (i.e., Phi-3.5-mini-instruct weights), an instruction-tuned model, and an enhanced version of its previous models in the Phi-3 category offered to the research community as a weights language model by Microsoft. The model was pre-trained on high-quality datasets available in the public domain and synthetic datasets. This version was enhanced with supervised fine-tuning, proximal policy optimization, and direct preference optimization to obtain good results with low-scale model parameters. It has a large token context window of 128K tokens using the LongRope [30] technique. The architecture of the Phi-3 is a decoder-only transformer, and each block is based on LLaMA 2 architecture. The complete details of the model parameters are presented in Table I.

C. Quantization Process

We also utilized quantized model weights in the evaluation framework to measure the impact of lower-precision model weights, such as the base benchmark weights, which are in half-precision using brain floating point (i.e., bfloat16). We used five different quantization processes: a 4-bit normalized floating point (NF4), NF4-dq for double quantization, a 4-bit floating point (FP4), FP4-dq for double quantization, and an 8-bit integer floating point (INT8). In Fig. 1, not quantized weights are depicted with full grids, while quantized weights are displayed with some grids off. These settings were configured using the bitsandbytes library [26]. Quantized models can help reduce device memory usage, making them suitable for mobile phones, Internet of Things, and edge devices.

IV. EXPERIMENTS

A. Datasets and Evaluation Metrics

Two datasets were used in our experiments to evaluate the selected SLM. The first dataset is ReCoRD [11], a large-scale reading comprehension dataset with commonsense reasoning characteristics, and the second is CosmosQA [12], a multi-choice question-answering dataset for commonsense reasoning collected from personal blogs on the web. ReCoRD is a cloze-style query in which the model must fill in suitable answers according to the provided context of the passage, and we converted it to a multi-choice question-answering setting for our needs. It is collected from news articles, focusing on complex query designs that normal MRC systems struggle to answer correctly and exclude too many complex queries that cannot be answered. In contrast, CosmosQA is a narrative understanding dataset in which the primary distribution of tasks is the cause and effect of events, facts about entities, and counterfactuals. The statistics of commonsense reasoning types for both datasets are presented in Table II. The sample sizes for the distribution of question types in the ReCoRD and CosmosQA datasets are randomly selected as 75 examples [11] and 500 examples [12], respectively. These question types are categorized manually for both datasets. We used the Exact Match (EM) and F1 Score for all experiments as the evaluation metrics.

TABLE II
DISTRIBUTION OF COMMONSENSE REASONING TYPES

Datasets	CSR Type	Percentage (%)
ReCoRD	Conceptual Knowledge	49.3
	Causal Reasoning	32.0
	Naıve Psychology	28.0
	Other	12.0
CosmosQA	Pre or Post Condition	27.2
	Motivation	16.0
	Reaction	13.2
	Temporal Events	12.4
	Situational Fact	23.8
	Counterfactual	4.4
	Other	12.6

B. Experimental Setup

We used the PyTorch Lightning framework [27], along with the Transformer [29] and Pytorch [28], for our experiments. The model weights are obtained from the Hugging Face and are accessible to all. For our hardware requirements, we used one Quadro RTX 6000 GPU. We utilized the bitsandbytes library for the model quantization process. Our codebase is configured for inference setup only, as our research involves evaluating the model's ability to follow instructions rather than fine-tuning; therefore, using only the generating feature of the autoregressive model is sufficient for our process. We used few essential parameters for answer-generation functionality, such as top-k, top-p, temperature, and maximum new tokens. The values specified for the top-k, top-p, temperature, and maximum new tokens are 50, 1.0, 0.8, and 50, respectively. Limiting the response of the model is very important for our task because the autoregressive model working principle is next token generation, so sometimes it is difficult to limit the verbose nature of decoder-only models; however, our required

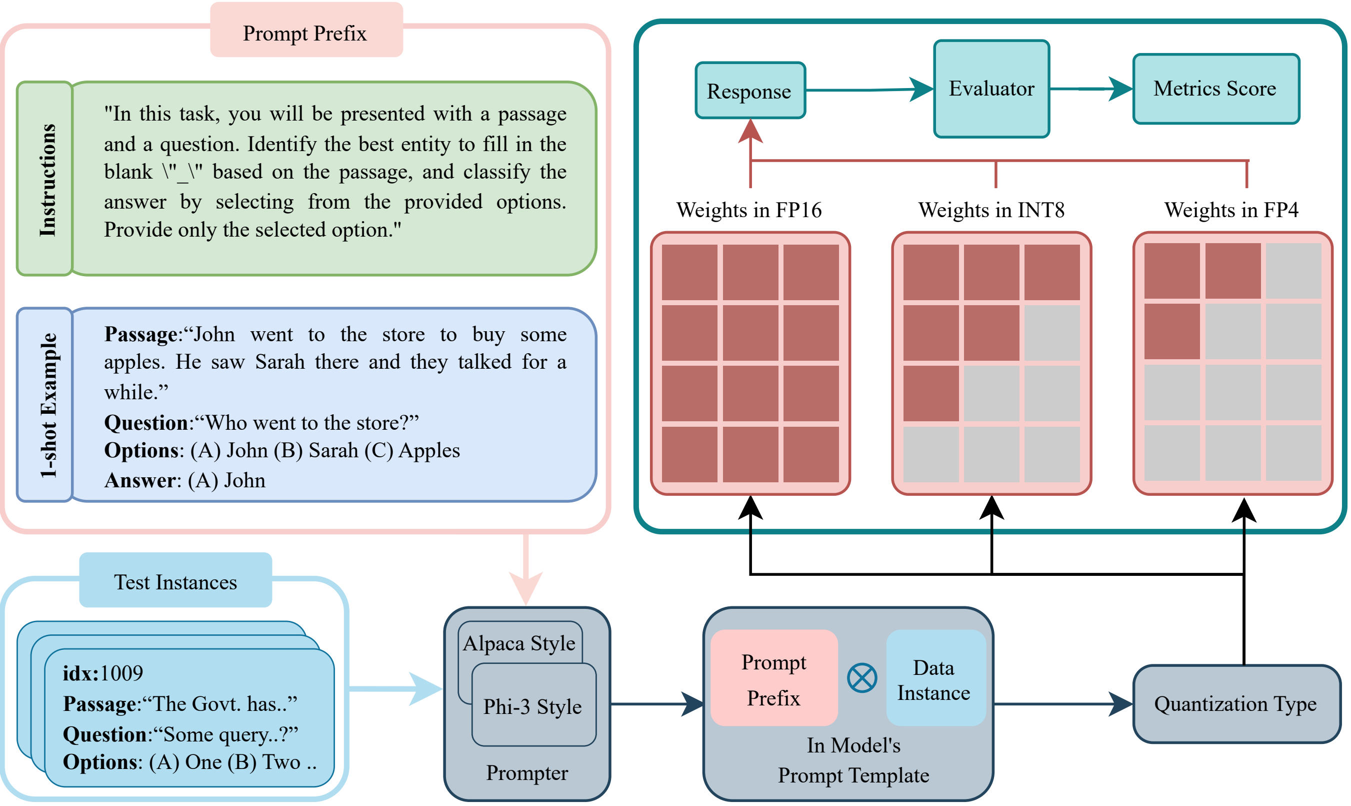
Prompt Prefix
Instructions
"In this task, you will be presented with a passage and a question. Identify the best entity to fill in the blank \"_\" based on the passage, and classify the answer by selecting from the provided options. Provide only the selected option."
1-shot Example
Passage:"John went to the store to buy some apples. He saw Sarah there and they talked for a while."
Question:"Who went to the store?"
Options: (A) John (B) Sarah (C) Apples
Answer: (A) John
Response
Evaluator
Metrics Score
Weights in FP16
Weights in INT8
Weights in FP4
Test Instances
idx:1009
Passage:"The Govt. has.."
Question:"Some query..?"
Options: (A) One (B) Two ..
Alpaca Style
Phi-3 Style
Prompter
Prompt Prefix
⊗
Data Instance
In Model's Prompt Template
Quantization Type

Fig. 1. Overview of the methodology we used to evaluate SLM for commonsense reasoning

answers are not very long, so we specify this requirement in our prompt instruction so that the model can predict only the necessary tokens.

V. Results and Analysis

A. Results

This section presents the findings from the one-shot evaluation of SLM on the ReCoRD and CosmosQA datasets. Using instruction prompting, various model configurations were tested to assess the impact of precision formats and quantization techniques on performance metrics such as EM, F1 Score, and Accuracy.

1) ReCoRD Dataset Results: The performance of different precision and quantization combinations on the ReCoRD dataset is summarized in Table III. Two key evaluation metrics—EM and F1 Score—were used to measure model performance. The model achieved an EM of 28.30% and an F1 Score of 30.06% without quantization. With NF4 and NF4-dq quantization, the scores slightly dropped, but remained close to the unquantized model's performance. The FP4 and FP4-dq configurations, however, showed a more pronounced decline in both EM (24.24% and 24.15%) and F1 Score (26.35% and 26.37%), respectively. Using the INT8 quantization method with float16 precision, the model demonstrated a significant boost, reaching an EM of 43.60% and an F1 Score of 44.70%, clearly outperforming all quantization configurations with bfloat16.

TABLE III
One-shot Evaluation Results for ReCoRD dataset

Model Configration		Evaluation Metrics	
Precision	Quantization	Exact Match (%)	F1 Score (%)
	None	28.30	30.06
	NF4[a]	27.23	29.75
bfloat16[e]	NF4-dq[b]	27.26	29.60
	FP4[c]	24.24	26.35
	FP4-dq	24.15	26.37
float16	INT8[d]	43.60	44.70

[a]4-bit Normalized Float [b]dq is Double Quantization [c]4-bit Float [d]8-bit Integer [e]16-bit BrainFloat

2) CosmosQA Dataset Results: Table IV presents the results of various precision and quantization techniques evaluated on the CosmosQA dataset, where the primary metric was Accuracy. The model attained an accuracy of 60.30% without quantization. Similar to the ReCoRD dataset, the NF4 and NF4-dq quantization methods resulted in only minor fluctuations, with NF4-dq slightly improving accuracy (60.34%). However, the FP4 and FP4-dq configurations saw a decline in accuracy, reaching 55.31% and 55.04%, respectively. The

TABLE IV
One-shot Evaluation Results for CosmosQA dataset

Model Configration		Evaluation Metrics
Precision	Quantization	Accuracy (%)
bfloat16[e]	None	60.30
	NF4[a]	59.40
	NF4-dq[b]	60.34
	FP4	55.31
	FP4-dq	55.04
float16	INT8[d]	63.79

[a] 4-bit Normalized Float [b] dq is Double Quantization [c] 4-bit Float [d] 8-bit Integer [e] 16-bit BrainFloat

highest accuracy (63.79%) was achieved using float16 precision with INT8 quantization, again outperforming all other configurations.

The results show that quantization can significantly impact model performance, with INT8 quantization in float16 precision yielding the best results across both datasets. Meanwhile, FP4 and FP4-dq quantization in bfloat16 consistently led to decreased performance. These findings suggest that lower-precision formats can be detrimental unless combined with efficient quantization techniques, such as INT8, which maximizes performance.

B. Analysis and Discussion

The line graph in Fig. 2 presents a comparative analysis of the model performance across two different precision types and various quantization methods for both ReCoRD and CosmosQA datasets. The performance was measured in terms of EM and F1 Score for ReCoRD and accuracy for CosmosQA. The trend line in this graph shows the results for both datasets, revealing that INT8 quantization consistently yields better results, whereas FP4-based quantization methods result in a significant performance decrease. These findings support the idea that INT8 quantization is the best option for maintaining performance in low-precision models with slightly more memory consumption trade-offs, particularly for commonsense reasoning tasks.

The graph in Fig. 3 presents a benchmarking analysis of inference speed (tokens per second) and GPU memory usage across different quantization methods (i.e., None is for not quantized, NF4, NF4-dq, FP4, FP4-dq, and INT8). Graph trend line represents the trade-offs between inference speed and memory consumption across all quantization methods. This comparison indicates that INT8 quantization with float16 precision is the most efficient method, as it performed the highest token generation rate with a little higher memory usage than other quantization methods. On the other hand, FP4 and FP4-dq are the slowest in token generation speed but most efficient for GPU memory consumption. Finally, NF4 and NF4-dq quantization provides a balanced performance compared to the previous two methods for a better trade-off between token generation speed and GPU memory allocation.

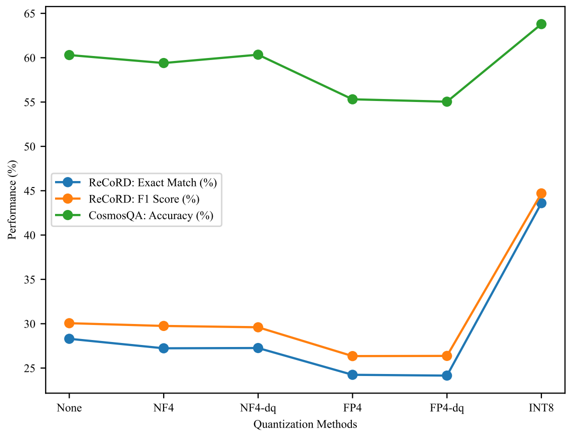

Fig. 2. Performance comparison across Quantization methods

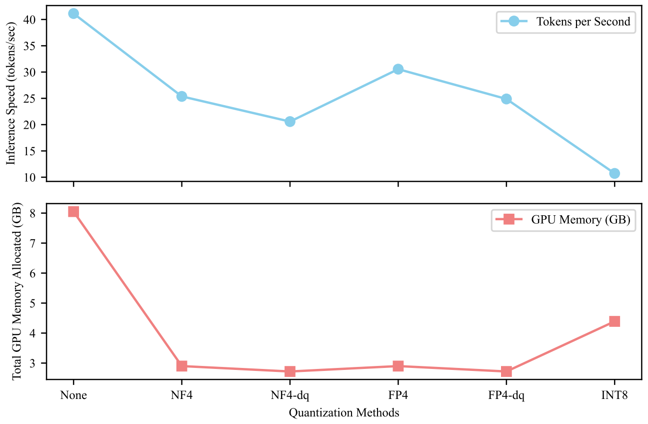

Fig. 3. Benchmarking tokens generation and GPU memory usage across different Quantization methods

VI. Conclusion and Future Work

This study evaluated Small Language Models (SLMs), specifically Phi-3, on commonsense reasoning tasks using instruction tuning-based evaluation. Different quantization techniques have also been used to evaluate the performance of SLMs with lower precision points. SLMs can effectively follow instructions and perform well on commonsense reasoning tasks without fine-tuning. The quantization method significantly impacts the performance, with 8-bit integer quantization outperforming the others. Trade-offs exist between the model performance, inference speed, and memory usage. The implications for deploying SLMs in resource-constrained environments are also discussed. Our study was limited to two datasets. Future research will expand to a broader range of commonsense reasoning and MRC datasets to comprehensively evaluate the SLM capabilities. While our study focused on instruction-tuning evaluation for both standard and quantized weights, future work will investigate the impact of fine-tuning and quantized fine-tuning of SLMs on target datasets to enhance their commonsense reasoning capabilities.

References

[1] E. Davis and G. Marcus, 'Commonsense reasoning and commonsense knowledge in artificial intelligence', Commun. ACM, vol. 58, no. 9, pp. 92–103, 2015.

[2] A. Wang et al., 'SuperGLUE: a stickier benchmark for general-purpose language understanding systems', in Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc., 2019.

[3] A. Vaswani et al., "Attention is all you need," in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[4] M. Abdin et al., "Phi-3 technical report: A highly capable language model locally on your phone," arXiv [cs.CL], 2024.

[5] T. Brown et al., 'Language Models are Few-Shot Learners', in Advances in Neural Information Processing Systems, 2020, vol. 33, pp. 1877–1901.

[6] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in Proceedings of the 2019 Conference of the North, 2019, pp. 4149–4158.

[7] P. Kavumba et al., "When Choosing Plausible Alternatives, Clever Hans can be Clever," in Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, 2019, pp. 33–42.

[8] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "WinoGrande: An adversarial winograd schema challenge at scale," Commun. ACM, vol. 64, no. 9, pp. 99–106, 2021.

[9] Y. Bisk, R. Zellers, R. Le bras, J. Gao, and Y. Choi, "PIQA: Reasoning about physical commonsense in natural language," Proc. Conf. AAAI Artif. Intell., vol. 34, no. 05, pp. 7432–7439, 2020.

[10] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Social IQa: Commonsense Reasoning about Social Interactions," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 4463–4473.

[11] Z. Sheng et al., "ReCoRD: Bridging the gap between human and machine commonsense reading comprehension," arXiv [cs.CL], 2018.

[12] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi, "Cosmos QA: Machine reading comprehension with contextual commonsense reasoning," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2391–2401.

[13] M. E. Peters et al., "Deep contextualized word representations," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in Proceedings of the 2019 Conference of the North, 2019, pp. 4171–4186.

[15] Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*. **21** (2020,1)

[16] Wei, J. et al., Finetuned Language Models are Zero-Shot Learners. *International Conference On Learning Representations*.

[17] H. Touvron et al., LLaMA: Open and Efficient Foundation Language Models', ArXiv, vol. abs/2302.13971, 2023.

[18] S. Biderman et al., Pythia: a suite for analyzing large language models across training and scaling, in Proceedings of the 40th International Conference on Machine Learning, 2023.

[19] M. Bellagente et al., 'Stable LM 2 1.6B Technical Report', ArXiv, vol. abs/2402.17834, 2024.

[20] G. T. M. Riviere et al., 'Gemma 2: Improving Open Language Models at a Practical Size', ArXiv, vol. abs/2408.00118, 2024.

[21] S. Longpre et al., The flan collection: designing data and methods for effective instruction tuning, in Proceedings of the 40th International Conference on Machine Learning, 2023.

[22] Y. Wang et al., 'Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks', in Conference on Empirical Methods in Natural Language Processing, 2022.

[23] Ye, Q., Lin, B. & Ren, X. CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP. *Proceedings Of The 2021 Conference On Empirical Methods In Natural Language Processing*. pp. 7163-7189 (2021,11).

[24] S. Bach et al., 'PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts', in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2022, pp. 93–104.

[25] L. Ouyang et al., 'Training language models to follow instructions with human feedback', in Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 27730–27744.

[26] Dettmers, T. & Zettlemoyer, L. The case for 4-bit precision: k-bit inference scaling laws. *International Conference On Machine Learning*. pp. 7750-7774 (2023).

[27] AI, L. LitGPT. (https://github.com/Lightning-AI/litgpt,2023)

[28] A. Paszke et al., 'PyTorch: an imperative style, high-performance deep learning library', in Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc., 2019.

[29] T. Wolf et al., 'HuggingFace's Transformers: State-of-the-art Natural Language Processing', ArXiv, vol. abs/1910.03771, 2019.

[30] Y. Ding et al., "LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens.," in ICML, 2024.