Meta-LLaMA: An end-to-end framework for data generation and meta-training.

Sunil Sabnis
Cornell Tech
New York, NY
svs57@cornell.edu

Ethan Joseph
Cornell Tech
New York, NY
eaj73@cornell.edu

Abstract

In this paper, we introduce Meta-LLaMA, a novel framework that combines the techniques of SelfInstruct and Meta-ICL to improve the few-shot learning (FSL) capabilities of large language models (LLMs). LLMs have shown remarkable performance in various natural language processing tasks but still exhibit limitations in FSL, which hinders their generalizability. We attempt to address this problem by meta-training LLMs with augmented data generated using a data generation method inspired by SelfInstruct. Our framework consists of two main stages: data generation and meta-training. We evaluate our framework on the BIG-Bench benchmark, a collection of diverse and challenging tasks, and compare the performance of our meta-trained models with baseline models. Our experimental results demonstrate that Meta-LLaMA slightly improves the few-shot learning performance of billion parameter LLMs, however they also suggest that the unique and varied characteristics exhibited by BIG-bench tasks result in the models acquiring a limited set of transferable skills during meta-training. We believe that our framework offers a promising approach to enhance generate additional data for diverse tasks, and could be a useful tool in constructing large datasets of diverse tasks without significantly less cost.

1 Introduction

Large language models (LLMs) have sparked serious conversations on what the future of work, society, and life might look like. (Eloundou et al., 2023) envision a future that is largely impacted by LLMs. One impressive property of LLMs is few-shot learning: the property of learning from examples to adapt to a new task without any gradient updates to the model's weights (Brown et al., 2020). Few-shot learning affords stronger generalizability and a powerful way to interact with LLMs. It can be viewed as a subset of meta-learning, a subfield of machine learning concerned with learning to learn and improving a model's generalizability (yi Lee et al., 2022).

The recent release of the LLaMa model family has closed the performance gap between open-sourced and closed-sourced models (Touvron et al., 2023). Open-sourced models fine-tuned with instruction style data are competitive with state-of-the-art closed-sourced models (Taori et al., 2023; Chiang et al., 2023). The instruction data can be generated from another LM (e.g., GPT-3) and instruction-tuning can be seen as a self-distillation process. While exhibiting strong dialogue behavior, these instruction-tuned models still have lackluster few-shot learning performance.

Similar to the problem of models hallucinating outputs (e.g., producing generations that are not true), state-of-the-art models often produce outputs that seemingly disregard in-context examples (Lu et al., 2022). Our study bridges ideas from (Wang et al., 2022) and (Min et al., 2022) by meta-training LLMs with augmented data to improve few-shot learning performance.

Our contributions include the following:

1. A novel data generation method for augmenting few-shot learning datasets.

2. A novel meta-training gradient update method that allows for support of a wider range of in-context examples.

3. Meta-training and evaluation of 7 billion parameter language models on a dataset of extremely diverse tasks.

We make our code available on GitHub.[1]

2 Related Works

Meta Learning for In-Context Learning

Recent work in meta-learning and in-context learn-

[1] https://github.com/sirmammingtonham/meta-llama

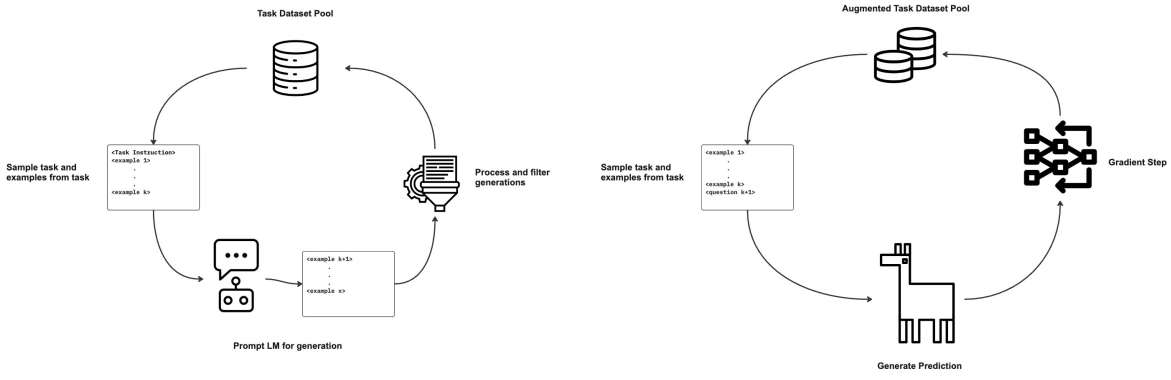

Figure 1: Data Generation Process (left) and Meta-Training Method (right)

ing have proposed methods to improve the generalizability of language models (LMs) (Kirsch et al., 2022; Min et al., 2022; Chen et al., 2022; Deb et al., 2022). (Min et al., 2022) proposed meta-training for in context learning (Meta-ICL), a training framework to improve in-context learning (ICL) capabilities of language models. Meta-ICL significantly improved the few-shot learning performance of GPT-2 large. Unlike (Min et al., 2022), however, our study applies a meta-training framework to a more challenging and diverse task data and we train a stronger and larger foundational model, LLaMa-7B. Similarly, (Chen et al., 2022) proposed in-context tuning, another meta-learning approach that fine-tunes few-shot learning with instructions provided within the input (prompt). However, their study also experiments with relatively small models (<1B parameters) and only concerns entity prediction and binary classification tasks. In this study, we experiment with highly varied task types.

Instruction-Tuning Language Models

Instruction-based data are tasks with natural language instruction descriptions. Fine-tuning language models on instruction-based data has been shown to improve zero-shot and few-shot learning (e.g., FLAN). On certain tasks, FLAN-137B in a zero-shot setting even outperforms GPT3-175B in a few-shot setting (Wei et al., 2022). Training FLAN, however, required careful dataset curation. In this study, we augment an initial small collection of tasks by artificially generating additional instruction-style examples using a LM. (Wei et al., 2022) focused on improving zero-shot learning, while our study concerns strengthening few-shot learning. One limitation of instruction-based data is that collecting the data is a costly, human-intensive process. (Wang et al., 2022), however, proposed the self-instruct framework: a method to artificially generate more instruction data samples. Models finetuned with data generated following self-instruct exhibit impressive zero-shot performance (Taori et al., 2023; Wang et al., 2022). In this study, we leverage a similar data generation process, but instead of prompting a language model to generate more instruction instances, the model is prompted to produce additional in-context examples. While (Wang et al., 2022; Taori et al., 2023) focus on improving alignment, we aim to improve few-shot learning.

Fine-Tuned LLaMa Models

The release of the LLaMa models has enabled the rapid development of open-source models competitive with state-of-the-art, closed-source models (Chiang et al., 2023; Taori et al., 2023; Geng et al., 2023). Remarkably, this performance is achieved with a small amount of data and compute. Currently, however, fine-tuned LLaMa models focus on instruction-tuning and optimizing for behavior similar to ChatGPT. These models display underwhelming few-shot learning capabilities. With this work, we aim to address this problem with a few-shot training objective.

3 Background

BIG-Bench

The beyond the imitation game benchmark (BIG-bench) is a collection of diverse, challenging tasks intended to benchmark the capabilities of LLMs. Tasks vary from math and computer science to social bias and common-sense reasoning. For example, the penguins in a table task prompts requires answering questions about a dataframe of penguins (e.g., average height of penguins in the table). BIG-bench provides a robust metric to measure the capabilities and limitations of current and future LLMs (Srivastava and others, 2022).

Meta-Training Formulation

Meta-training is a training paradigm for learning few-shot learning (learning by examples) (Min et al., 2022). Fine-tuning often requires training a model on a single task, T_i. A single training task, T_i is composed of n individual examples, $T_i = \{(x_0, y_0), (x_1, y_1)..., (x_n, y_n)\}$. During training, a model optimizes $P(y_i|x_i)$. However, for meta-training, we are given a collection of training tasks: $\mathbf{T} = \{T_0, T_1, ..., T_n\}$. During meta-training, we sample a training task, T_i, sample $k+1$ examples from T_i, and optimize for $P(y_{k+1}|(x_0, y_0), (x_1, y_1), ...(x_k, y_k), x_{k+1})$.

4 Method

We present Meta-LLaMA, a novel framework that incorporates the powerful techniques of SelfInstruct and Meta-ICL to generate augmented task data and then meta-train a model on those tasks. This integration allows for a end-to-end approach that tries to address the challenges of in-context learning. To visualize the workflow of Meta-LLaMA, refer to Figure 1, which provides a detailed diagram illustrating the various stages and interactions within the framework.

4.1 Data Generation

We follow a similar data generation process as (Wang et al., 2022) with the major difference being that we generate task examples instead of instructions. We have a collection of distinct training tasks, $\mathbf{T} = \{T_0, T_1, ..., T_n\}$. Each T_i is initialized with a seed of 50 examples: $|T_i| = 50; \forall i \in [0, n]$. $\mathbf{T}^G = \{T_0^G, T_1^G, ..., T_n^G\}$ is a corresponding collection of generated examples for each task, T_i. Initially $|T_i^G| = 0; \forall i \in [0, n]$.

All example instances from a task, T_i, are valid JSON; as such, we prompt the model to output examples in this same format. To generate examples, we select a task T_i and the corresponding generated task set, T_i^G. We then sample k examples from $T_i \cup T_i^G$ and construct our prompt, P, with the k examples appended to an instruction. The instruction, contains context about the goal of the task (e.g., "Answer questions about a table of penguins and their attributes"), a JSON schema to follow, and a statement to generate x examples. Thus the model is prompted to generate $x - k$ novel task-related examples. See Appendix C for an example prompt used for generation.

From the model's output, we filter out examples that are duplicates or do not follow the JSON schema (e.g., missing keys). After filtering, the examples are added into the generated task set, T_i^G. This generation process is executed for all training tasks until we have generated at least 200 example instances for each task; $|T_i^G| \geq 200; \forall i \in [0, n]$.

4.2 Our Meta-training procedure

We adopt a similar meta-training procedure as described in (Min et al., 2022). In each iteration, we sample a meta-training task and select $k+1$ training examples denoted as $(x_1, y_1), \ldots, (x_{k+1}, y_{k+1})$ from that task's training data, where x represents the input and y represents the corresponding answer/label. The model is trained by providing the concatenated sequence $(x_1, \ldots, x_{k+1})$ as input and training it to generate the corresponding labels $(y_1, \ldots, y_{k+1})$ using a negative log likelihood objective.

This approach simulates in-context learning during inference, where the first k examples act as training instances, and the last example $(k+1)$ is treated as the test example. Notably, our method differs from (Min et al., 2022) in that we compute the loss for each y_i, rather than solely for the final y_{k+1}. Consequently, the model becomes proficient in handling contexts with 1 to k in-context examples.

See Appendix D for full details on the training prompt.

4.3 Inference

During inference, the model is given k randomly sampled training examples $(x_1, y_1), \ldots, (x_k, y_k)$ as well as a test input x'. As in meta-training, the model takes a concatenation of $x_1, y_1, \ldots, x_k, y_k, x'$ as the input, and computes the probability of label y' over the model's vocabulary. The token with the maximum probability is

returned as the predicted label.

5 Experiment Setup

We trained and evaluated two models, LLaMA-7B (Touvron et al., 2023) and Vicuna-7B (Chiang et al., 2023), under three settings: baseline, bigbench, and augment, with meta-training applied in the latter two settings.

5.1 Data

For data generation, we augment 188 tasks from BIG-bench (Srivastava and others, 2022). Each task follows a multiple-choice question format. This subset of tasks are diverse and include many distinct domains (e.g., physics, understanding_fables). See Appendix A for the full list of train tasks that were augmented. We use OpenAI's GPT-3.5 chat endpoint for generation. We set temperature to 0.7 and presence_penalty to 0.8 to encourage diverse, task-related example generations. We then evaluate on a held-out test set of tasks that we randomly sampled from all BIG-bench tasks. See Appendix B for a full list of test tasks. The total API cost for generating our augmented dataset of around 20,000 examples was around $50 USD.

5.2 Model

We conducted experiments involving two models: LLaMA-7B (Touvron et al., 2023) and Vicuna-7B (Chiang et al., 2023), in three distinct settings. The selection of LLaMA was driven by our desire to assess the effectiveness of our method in enhancing the in-context learning capability of large language models with billions of parameters. Vicuna is an instruction-tuned LLaMA model, and we wanted to compare the in-context learning capabilities of it compared to base LLaMA.

In the first setting, referred to as the "baseline," we performed no training and solely evaluated the models on the test set without any fine-tuning, serving as a benchmark. In the second setting, named "bigbench," we applied fine-tuning using the meta-training formulation detailed in Section 4.2, utilizing the original bigbench dataset. Lastly, our third setting, termed "augment," also involved fine-tuning using meta-training, but this time on our augmented dataset described in Section 4.1.

To reduce the enormous memory requirements for training these billion parameter models, we follow the training procedure of the Alpaca-LoRA repo: each model was trained in int8 precision with low rank adaptation (LoRA) (Hu et al., 2021) applied to the attention query and value projection weight matrices. We fine tuned each model in the "bigbench" and "augment" settings for 3 epochs with an initial learning rate of 5e-5 and k=3 due to context length limitations for certain tasks. All models were trained on a single NVIDIA A6000.

6 Results

In this section, we interpret the table of results (Table 1) showcasing the performance of different models under various settings in the context of in-context learning. The results are averaged over five runs to account for the random sampling of in-context examples.

Each model is meta-trained with k=3 on the training tasks and evaluated on a set of test tasks. We chose k=3 since many tasks have long inputs, so even with LLaMA's 2048 token context length, k=3 on average allows us to fit the most examples in context. We average results for each task across 5 evaluations with different seeds since we randomly sample training examples to serve as context during inference (refer to Section 4.3 for details).

We observe that all models perform rather poorly on the test tasks, with an average accuracy of less than 12% across the board. The most interesting observation is that training on our augmented dataset slightly boosts the average accuracy of both LLaMA and Vicuna when compared to training on the original Bigbench dataset.

Vicuna performs worse when it is meta-trained than without any meta-training, but meta-training on the augmented dataset has a smaller effect on accuracy than meta-training on the original dataset.

LLaMA is slightly improved by meta-training on both the original dataset and the augmented one, with the best model being a 0.6% improvement over the baseline.

Despite these results, we observe that the best performance for individual tasks is spread out between almost all models and settings.

For a comprehensive overview of all tasks and their respective accuracies, please refer to the complete results table in Appendix A (see Appendix E).

7 Analysis

In this section, we analyze the experimental setup and results of Meta-LLaMa. We pose and answer five questions to gain insights into the effectiveness of the proposed framework.

setting	model	average*	crash blossom	english proverbs	irony identification	metaphor boolean	physical intuition	sentence ambiguity	winowhy
Baseline	LLaMA-7B	0.109	0.125	0.083	0.127	**0.14**	0.052	0.142	**0.136**
	Vicuna-7B	0.11	0.148	0.088	0.131	0.136	0.052	0.133	0.131
Bigbench	LLaMA-7B	0.11	0.1	**0.119**	**0.162**	0.124	0.065	0.121	0.121
	Vicuna-7B	0.101	0.086	0.065	0.128	0.136	0.047	0.133	0.129
Augment	LLaMA-7B	**0.115**	**0.175**	0.063	0.123	0.113	**0.088**	0.152	0.121
	Vicuna-7B	0.108	0.155	0.069	0.101	0.119	0.077	**0.16**	0.102

Table 1: Sample test set accuracy of each model in each data setting with k=3. Best result bolded. Results are averaged over 5 runs due to the random sampling of in-context examples.

*The average is calculated using all test set tasks. However, due to space constraints, certain tasks are excluded from the table. For the complete results table, please refer to Appendix E.

setting	model	average
Baseline, $k = 2$	LLaMA-7B	0.1093
Augment, $k = 2$	LLaMA-7B	0.1042

Table 2: Sample test set accuracy of best performing models with k=2.

How do billion parameter LLMs perform on BIG-Bench?

We observe that the 7 billion parameter versions of LLaMA and Vicuna perform quite poorly on the test set tasks that we sampled, with an average accuracy of less than 12%. Empirically this seems to be worse than the performance reported in Gopher's 7B parameter BIG-Bench evaluation (Rae and others, 2022) (they report an accuracy of around 50% on winowhy), however they evaluated with $k = 5$, a different prompt, and no meta-training.

Is meta-training effective in improving the performance of billion parameter LLMs?

We observe that the meta-training procedure was not very effective in improving the performance of 7B LLaMA or Vicuna on the BIG-Bench test tasks. While it slightly improved the average accuracy of LLaMA-7B, it actually decreased the average accuracy for Vicuna-7B. We hypothesize that due to the distinct and diverse nature among BIG-bench tasks, the model acquires only a limited set of transferable skills during meta-training that can be effectively employed during evaluation.

How does the data generation process affect the performance of Meta-LLaMA?

The data generation process plays a crucial role in the performance of Meta-LLaMA. We employ a novel data generation method inspired by Self-Instruct (Wang et al., 2022), where the model is prompted to produce additional in-context examples. As shown in Table 1, we find that this process effectively augments the training data and slightly improves the few-shot learning performance of Meta-LLaMA.

Does our gradient-update method help handle contexts with varying numbers of in-context examples?

We perform an additional experiment in Table 2, where we take our augmented LLaMA-7B model meta-trained with $k = 3$ on the augmented dataset and evaluate it with $k = 2$. As shown in the table, the performance of the augmented model decreased compared to the baseline, indicating that our gradient update of computing the label loss for each in-context example actually does not improve the model's ability to handle different numbers of in-context examples as we hypothesized.

What are the limitations of Meta-LLaMA?

While Meta-LLaMA demonstrates a slight improvement over the baseline for the BIG-bench benchmark, it has certain limitations. First, the data generation process relies on the underlying language model's ability to generate high-quality in-context examples. However, there is a possibility of generating incorrect or nonsensical examples, which can adversely affect the model's performance. Second, the meta-training procedure requires sampling multiple in-context examples, which can possibly affect the quality of the generations. Future work could explore methods to address these limitations and further improve the performance of Meta-LLaMA.

8 Conclusion

In this paper, we have introduced Meta-LLaMA, a novel framework that combines the SelfInstruct and Meta-ICL techniques to enhance the few-shot learning capabilities of large language models. Our framework incorporates a data generation method that effectively utilizes the language model's capacity to generate in-context examples, thereby significantly augmenting the training data. Additionally, we have proposed a meta-training procedure

that equips the model to handle varying contexts with different numbers of in-context examples. Experimental results on the BIG-bench benchmark demonstrate that Meta-LLaMA surpasses baseline models, although all models perform relatively poorly on the diverse test set of tasks from BIG-bench. Our findings suggest that the distinct and diverse characteristics exhibited by BIG-bench tasks limit the models' acquisition of transferable skills during meta-training. Consequently, these skills can only be applied to a limited extent during the evaluation phase. Nevertheless, we are optimistic that our framework offers a promising approach to generate additional data for diverse tasks and could prove valuable in constructing large datasets of diverse tasks at a significantly reduced cost compared to traditional, human-driven methods.

8.1 Future Work

While Meta-LLaMa shows signs of promising results, especially in dataset generation/augmentation, there are several avenues for future work. First, we plan to explore methods to mitigate the risk of generating incorrect or nonsensical examples during the data generation process. This could involve incorporating additional filtering mechanisms or using reinforcement learning techniques to guide the generation process. Second, we aim to investigate why the performance of our multi-example gradient update performed worse than a baseline without any meta-training. This could involve exploring techniques such as gradient-based meta-learning or parameter sharing across tasks. Overall, we believe this proposed framework can open up exciting avenues for future research into meta-training with augmented data.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Budhaditya Deb, Guoqing Zheng, and Ahmed Hassan Awadallah. 2022. Boosting natural language generation from instructions with meta-learning.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. 2022. General-purpose in-context learning by meta-learning transformers.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Metaicl: Learning to learn in context.

Jack W. Rae et al. 2022. Scaling language models: Methods, analysis insights from training gopher.

Aarohi Srivastava et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Hung yi Lee, Shang-Wen Li, and Ngoc Thang Vu. 2022. Meta learning for natural language processing: A survey.

A BIG-Bench Training Tasks

abstract-narrative-understanding,anachronisms,analytic-entailment,arithmetic,bbq-lite-json,causal-judgment,cause-and-effect,checkmate-in-one,code-line-description,color,common-morpheme,conceptual-combinations,contextual-parametric-knowledge-conflicts,crass-ai,cryobiology-spanish,cs-algorithms,date-understanding,disambiguation-qa,discourse-marker-prediction,dyck-languages,elementary-math-qa,emoji-movie,emojis-emotion-prediction,empirical-judgments,english-russian-proverbs,entailed-polarity,epistemic-reasoning,evaluating-information-essentiality,fact-checker,fantasy-reasoning,figure-of-speech-detection,general-knowledge,geometric-shapes,goal-step-wikihow,gre-reading-comprehension,hhh-alignment,hindu-knowledge,hinglish-toxicity,human-organs-senses,identify-math-theorems,identify-odd-metaphor,implicatures,implicit-relations,international-phonetic-alphabet-nli,intersect-geometry,navigate,nonsense-words-grammar,novel-concepts,odd-one-out,penguins-in-a-table,periodic-elements,persian-idioms,phrase-relatedness,physics,play-dialog-same-or-different,presuppositions-as-nli,question-selection,real-or-fake-text,reasoning-about-colored-objects,rhyming,riddle-sense,ruin-names,salient-translation-error-detection,similarities-abstraction,simple-ethical-questions,snarks,social-iqa,social-support,sports-understanding,strange-stories,strategyqa,suicide-risk,swahili-english-proverbs,symbol-interpretation,temporal-sequences,timedial,tracking-shuffled-objects,understanding-fables,unit-conversion,vitaminc-fact-verification,what-is-the-tao,which-wiki-edit

B BIG-Bench Test Tasks

crash-blossom,dark-humor-detection,english-proverbs,entailed-polarity-hindi,formal-fallacies-syllogisms-negation,hyperbaton,intent-recognition,irony-identification,kannada,key-value-maps,language-identification,logical-sequence,mathematical-induction,medical-questions-russian,metaphor-boolean,metaphor-understanding,movie-dialog-same-or-different,movie-recommendation,physical-intuition,sentence-ambiguity,swedish-to-german-proverbs,undo-permutation,unit-interpretation,winowhy

C Data Generation Prompts

See Fig. 2 for full data generation prompt.

D Training and Inference Prompts

See Fig. 3 for an example training/inference prompt.

E Full Results

See Table 3 for full test set results.

DESCRIPTION: Answer questions about a table of penguins and their attributes.
SCHEMA:
{ "$schema": "http://json-schema.org/draft-04/schema#", "type": "object", "properties": { "inputs": { "type": "string" }, "targets": { "type": "array", "items": { "type": "string" } }, "multiple_choice_targets": { "type": "array", "items": { "type": "string" } } }, "required": ["inputs", "targets", "multiple_choice_targets"] }
Generate a series of {x} diverse questions related to DESCRIPTION in a valid JSON format that follows SCHEMA.
Question 0:
{ "inputs": "Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. We now add a penguin to the table: James, 12, 90, 12 And here is a similar table, but listing giraffes: name, age, height (cm), weight (kg) Jody, 5, 430, 620 Gladys, 10, 420, 590 Marian, 2, 310, 410 Donna, 9, 440, 650 What is the name of the last animal? Answer:", "targets": ["Donna"], "multiple_choice_targets": ["Bernard", "Donna", "Gladys", "Gwen", "James"] }
Question 1:
{ "inputs": "Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. And here is a similar table, but listing giraffes: name, age, height (cm), weight (kg) Jody, 5, 430, 620 Gladys, 10, 420, 590 Marian, 2, 310, 410 Donna, 9, 440, 650 How many penguins are there in the tables? Answer:", "targets": ["4"], "multiple_choice_targets": ["1", "2", "3", "4", "5"] }
Question 2:
{ "inputs": "Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. And here is a similar table, but listing giraffes: name, age, height (cm), weight (kg) Jody, 5, 430, 620 Gladys, 10, 420, 590 Marian, 2, 310, 410 Donna, 9, 440, 650 What is the cumulated age of the giraffes? Answer:", "targets": ["26"], "multiple_choice_targets": ["26", "29", "41", "55", "67"] }
Question 3:
{ "inputs": "Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. We now add a penguin to the table: James, 12, 90, 12 And here is a similar table, but listing giraffes: name, age, height (cm), weight (kg) Jody, 5, 430, 620 Gladys, 10, 420, 590 Marian, 2, 310, 410 Donna, 9, 440, 650 What is the name of the last giraffe? Answer:", "targets": ["Donna"], "multiple_choice_targets": ["Jody", "Gladys", "Marian", "Donna", "Louise"] }
Question 4:

Figure 2: Example Data Generation Prompt. $x - k$ with ($k = 4$) is the total amount of questions to generate.

Find a movie similar to Batman, The Mask, The Fugitive, Pretty Woman:
choice 0: The Front Page
choice 1: Maelstrom
choice 2: The Lion King
choice 3: Lamerica
answer: 2
Find a movie similar to The Sixth Sense, The Matrix, Forrest Gump, The Shawshank Redemption:
choice 0: Street Fighter II The Animated Movie
choice 1: The Sheltering Sky
choice 2: The Boy Who Could Fly
choice 3: Terminator 2 Judgment Day
answer: 3
Find a movie similar to Schindler's List, Braveheart, The Silence of the Lambs, Tombstone:
choice 0: Orlando
choice 1: Guilty of Romance
choice 2: Forrest Gump
choice 3: All the Real Girls
answer: 2

Figure 3: Example training/inference prompt with $k = 2$ on movie recommendation task.

setting	Baseline		Bigbench		Augment	
model	LLaMA-7B	Vicuna-7B	LLaMA-7B	Vicuna-7B	LLaMA-7B	Vicuna-7B
average	0.1093	0.1097	0.1099	0.1015	0.1151	0.108
crash blossom	0.125	0.1484	0.0995	0.0857	0.175	0.155
dark humor detection	0.129	0.1484	0.1355	0.1476	0.1597	0.1419
english proverbs	0.0825	0.0877	0.1187	0.0649	0.0625	0.0688
hyperbaton	0.1309	0.1329	0.1117	0.1266	0.1153	0.1274
irony identification	0.1269	0.1308	0.1615	0.1278	0.1231	0.1013
mathematical induction	0.1549	0.1353	0.1431	0.1346	0.149	0.1333
metaphor boolean	0.1399	0.1359	0.1238	0.1361	0.1133	0.1194
movie recommendation	0.0577	0.0573	0.0649	0.0562	0.0646	0.0557
physical intuition	0.0524	0.0524	0.0651	0.0469	0.0877	0.0766
sentence ambiguity	0.142	0.1325	0.1214	0.1331	0.1524	0.1595
swedish to german proverbs	0.087	0.0926	0.1019	0.0782	0.1111	0.1037
unit interpretation	0.0564	0.041	0.0603	0.0519	0.0615	0.059
winowhy	0.1357	0.1313	0.121	0.1294	0.121	0.1024

Table 3: Sample test set accuracy of each model in each data setting with k=3 for each task in the test set.