
LLaMa-SciQ: An Educational Chatbot for Answering Science MCQ

Marc-Antoine Allard¹, Matin Ansari pour¹, Maria Yuffa¹, Paul Teiletche¹

¹EPFL, Lausanne, The Wordsmiths
{firstname.lastname}@epfl.ch

Abstract

Large Language Models (LLMs) often struggle with tasks requiring mathematical reasoning, particularly multiple-choice questions (MCQs). To address this issue, we developed LLaMa-SciQ, an educational chatbot designed to assist college students in solving and understanding MCQs in STEM fields. We begin by fine-tuning and aligning the models to human preferences. After comparing the performance of Mistral-7B and LLaMa-8B, we selected the latter as the base model due to its higher evaluation accuracy. To further enhance accuracy, we implement Retrieval-Augmented Generation (RAG) and apply quantization to compress the model, reducing inference time and increasing accessibility for students. For mathematical reasoning, LLaMa-SciQ achieved 74.5% accuracy on the GSM8k dataset and 30% on the MATH dataset. However, RAG does not improve performance and even reduces it, likely due to retriever issues or the model's unfamiliarity with context. Despite this, the quantized model shows only a 5% loss in performance, demonstrating significant efficiency improvements.

1 Introduction

Large Language Models (LLMs) are known to perform poorly on questions requiring advanced mathematical reasoning (Wu et al., 2023). This is especially true for the university level problems (Wang et al., 2024). In literature, the failure of current approaches is attributed to inability of LLMs to recognize and correct a wrong answer (Imani et al., 2023) as well as catastrophic forgetting of linguistic skills when trained on maths data (Sharma et al., 2023). The issues cannot be fully addressed with a simple prompting strategy due to data variability (Wang et al., 2024).

This project explores state-of-the-art LLMs for creation of an accessible chatbot that assists students in mathematics, physics and computer science. Specifically, we fine-tune LLaMa-3-8B

model as well as Mistral-7B on a variety of mathematical and scientific datasets further using Direct Preference Optimization (DPO) to align model's responses to the ones preferred by a student. We compare the performances of the models and demonstrate the significantly superior performance of the LLaMa model with which we proceed. We try to enhance the accuracy of fine-tuned LLaMa-3-8B model by applying Retrieval Augmented Generation (RAG). Finally, we quantize the LLM for more efficient inference, making it suitable for students needs.

2 Related Work

With recent release of ChatGPT-3.5 and ChatGPT-4, the number of people using LLMs for education has sky-rocketed (Fütterer et al., 2023). To leverage its capabilities while making user-friendly interfaces vast amount of research is dedicated to creation of LLM based Chatbots for academic purposes (Odede and Frommholz, 2024). Despite success of ChatGPT models on linguistic tasks, their performance was limited on problems involving mathematical reasoning. This is especially true for MCQ questions where the answer is not verbal. This is showcased by the work of (Savelka et al., 2023), where GPT model struggled to give the correct answer to the questions that do not contain natural language.

To improve the performance of the pre-trained LLM model on mathematical questions while ensuring the alignment of the responses with the intended purposes and human values, we considered both Supervised Fine-tuning on mathematical and scientific datasets as well as DPO on the preference pairs ranked by students. This approach was inspired by InstructGPT (Ouyang et al., 2022) in aligning LLMs with human preferences. We also considered DPO with an offset (Amini et al., 2024). This approach introduces variability in treatment of preference pairs and could be less robust, since

the inferred offset value might be high in a mis-annotated responses and confuse the model. Therefore, to achieve good results on noisy data while keeping the implementation simple, we chose Conservative DPO (cDPO) loss for DPO fine-tuning strategy, primarily due to its robustness on noisy data.

Further refining the model, we have considered RAG. Initially, we aimed to use pre-trained RAG retriever (Karpukhin et al., 2020) or adopting the novel concept of Retrieval Augmented Fine-Tuning (RAFT) (Zhang et al., 2024). Thoroughly reviewing the literature (Gao et al., 2024) we yielded to the *Naive RAG* strategy due to good performance and straightforwardness of the approach.

Finally, we considered quantizing the model to reduce the computational costs when using the Chatbot while maintaining good response accuracy. At first, we sought to use QuIP (Chee et al., 2024) and recently released QuIP# (Tseng et al., 2024) due to its ability to leverage incoherence in weights and Hessian matrices. We have also considered QLoRA (Hu et al., 2021), which is not computationally demanding, yet preserves the 16-bit fine-tuning task performance of LLMs. Before attempting the advanced methods, we have tried GPTQ (Frantar et al., 2023). Nonetheless, facing some numerical issues with quantizing our model with GPTQ, we decided to use the 4-bit quantization provided by Unsloth bitsandbytes library.

Overall, our work is a nice step towards creating an efficient, student-oriented educational assistant for questions requiring mathematical reasoning.

3 Approach

Our approach consisted of performing SFT training on both Mistral-7B and LLaMa-3-8B. We then compared the performance of two models with SFT and DPO training and proceeded with LLaMa-3-8B which performed better on the evaluation set (Figure 1). In this section, we outline the details of the models and their fine-tuning strategy with emphasis on LLaMa-3-8B.

3.1 Base Model Architecture

LLaMa-3-8B (AI@Meta, 2024) is an autoregressive language model featuring an enhanced transformer architecture with a standard decoder-only design. The model integrates supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to better align with hu-

man preferences regarding safety and helpfulness. Llama 3, which uses a tokenizer with a 128K-token vocabulary for more efficient language encoding, shows significant performance improvements over its predecessor. The model, trained on sequences up to 8,192 tokens with boundary-aware self-attention, uses Grouped-Query Attention (GQA) to enhance inference scalability.

Mistral-7B (Jiang et al., 2023), a language model with 7 billion parameters, utilizes a transformer-based architecture comprising multiple transformer blocks. It employs sliding window attention that allows the model to attend to tokens outside of the window, Rolling Buffer Cache to reduce the cache memory usage while keeping the model quality. It also utilizes pre-fill and chunking, which involve loading known parts of a prompt into the (k, v) cache to facilitate token generation. If the prompt is lengthy, it is segmented into smaller chunks, each pre-filled into the cache to enhance processing efficiency during token prediction.

3.2 Training Pipeline

The training pipeline for LLaMa-3-8B model is demonstrated in Figure 1. We first performed Supervised Fine-tuning on a mix of specialized maths and science datasets. We then performed DPO training using preference data generated and annotated by students via cDPO loss. Finally, we gauged the performance of the model on AQUA-Rat (Ling et al., 2017) dataset which contains STEM-related MCQ questions.

Mistral-7B used the same process as LLaMa, except for the final SFT, since LLaMa showed superior performance (Figure 1). Ultimately, we have not implemented both due to time constraints.

3.2.1 Supervised Fine Tuning

The results of supervised fine-tuning of two models are demonstrated in Table 1.

3.2.2 Preference Data Collection

To collect preference data, a cohort of 300 students was asked to generate two responses, a better one and a slightly worse one but preferably still correct, to the question using GPT-wrapper. The students were further asked to rank the responses.

To generate answers for the dataset of questions in mathematics, physics, and computer science, we have developed a prompting strategy that incorporates several techniques. Firstly, we create a

separate chat for each subject id. Secondly, we use Chain-of-Thought (CoT) (Wei et al., 2022), which guides the model to reach conclusions in a step-by-step manner. Thirdly, the model is prompted with the instruction provided in B.5. Finally, for generating preference pairs, we employ the following method: to achieve a better response, we prompt the model to re-read the question before attempting to solve it. This method was shown by (Xu et al., 2024) to consistently improve performance for LLMs, except for Vanilla ChatGPT. For the worst answer, the model is instructed to provide a very brief explanation.

3.2.3 Reward Model

The reward model is a critical component of the DPO fine-tuning strategy. The reward model is based on the policy that maximizes the reward with KL constraint to the reference policy:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{y \sim \pi} \left[r(y) - \beta \log \frac{\pi(y)}{\pi_{\text{ref}}(y)} \right]$$

Considering a small probability that the preference pair could be flipped, the preferred response is in reality less correct or explicit than the other one, we can derive the following DPO loss to optimise the reward model:

$$\mathcal{L}_{DPO}^{\epsilon}(\theta, y_w, y_l) = -(1 - \epsilon) \log \hat{p}_{\theta}(y_w > y_l) \quad (1)$$

$$- \epsilon \log(1 - \hat{p}_{\theta}(y_w > y_l)) \quad (2)$$

$$= (1 - \epsilon) \mathcal{L}_{DPO}(\theta, y_w, y_l) \quad (3)$$

$$+ \epsilon \mathcal{L}_{DPO}(\theta, y_l, y_w), \quad (4)$$

where ϵ indicates the probability of the answer being wrong (or flipping the pair).

Base Model	Strategy	Test Rwrds Acc.
LLaMa-3-8B	DPO	79.7%
LLaMa-3-8B	SFT+DPO	79.3%
Mistral-7B	DPO	76.5%
Mistral-7B	SFT+DPO	71.3%

Table 1: Accuracy scores of the models on 1000 samples of the test set.

3.3 Retrieval Augmented Generation

We augment LLaMa-SciQ by incorporating Retrieval-Augmented Generation (RAG) methods.

This approach stands out as one of the most effective means to enhance the predictive capabilities of our model. RAG combines the capabilities of generative models, dense vector indices of a corpus of documents, and pre-trained neural retrievers. Figure 2 summarizes our RAG pipeline, known as the Naive RAG. We use the dataset described in 4.1.4 as our Dense Passage Retrieval (DPR) corpus of documents over which we create an index using Facebook’s FAISS library (Douze et al., 2024). Documents are then retrieved using Facebook’s DPR question encoder (Karpukhin et al., 2020) and added to the prompt in the format detailed in appendix D.1.

We observed that our model is getting biased by saying to use the provided information. So we changed the prompt for RAG to tell the model to consider the model but not get biased on the information and try to fulfill the objective of the questions.

3.4 Quantization

We took an alternative route compared to standard quantization techniques. In particular, we specified the bytes-and-bits (bnb) parameter when loading the model using Unloth package. We, therefore, reduced the weights to 4-bits while sustaining the accuracy.

When you enable "load_in_4bits" in the "from_pretrained" function of the unloth repository, the model utilizes a quantization technique facilitated by the bitsandbytes library. This technique allows the model weights to be represented with 4-bit precision, significantly reducing the model’s memory footprint while attempting to preserve performance.

This 4-bit quantization primarily involves the transformation of model weights, previously in full-precision formats like fp16 or bf16, into a 4-bit format. The process entails creating instances of linear layers designed for 4-bit operations (e.g., "Linear4bit"), and then loading the original model’s weights into these quantized modules. The actual quantization happens when these modified models are transferred to a computation device like a GPU.

This quantization approach can utilize different data types for quantization, like FP4 (Float4) or NF4 (NormalFloat4), which are tailored for different kinds of data distributions and usage scenarios. For example, NF4 is designed for data that naturally follows a normal distribution, offering

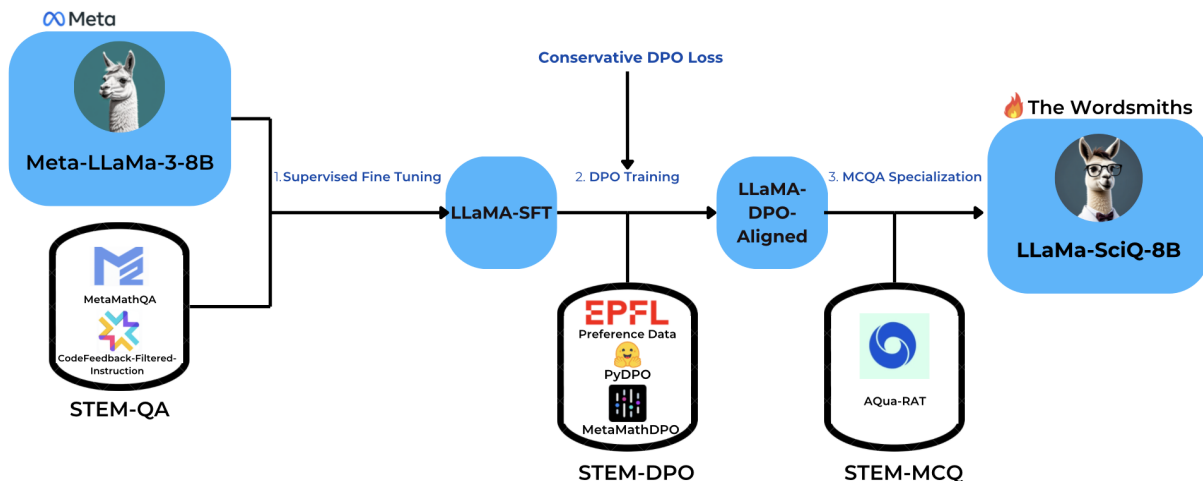


Figure 1: The Training Pipeline: Organized into three consecutive stages; Supervised Fine-Tuning, Direct Preference Optimization Training, and Multiple Choice Question Answering Specialization.

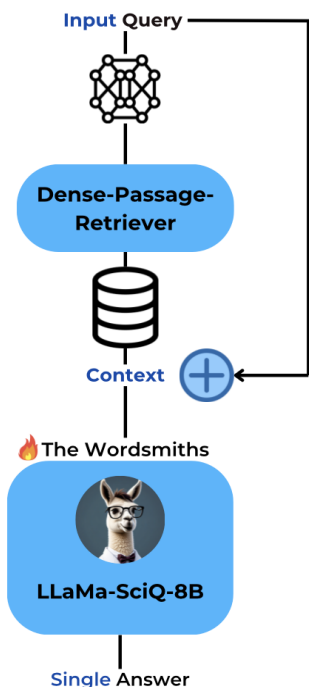


Figure 2: The RAG Pipeline

potential performance improvements in such cases.

4 Experiments

4.1 Data

This section outlines the datasets created for our model’s alignment stages. Samples of the datasets can be found in Appendix B.

4.1.1 SFT Dataset

We first introduce StemQA, a specialized dataset to extend our model’s performance on math and coding questions. This dataset is a

blend of MetaMathQA (Yu et al., 2023) and CodeFeedback-Filtered-Instruction (Zheng et al., 2024) datasets. It is balanced so that 75% of the questions are math-related, while the remaining 25% are coding-related. Table 2 presents these proportions. The answers now include the rationale followed by "The answer is: <Maths/Code>" to simplify future answer extraction.

Dataset	Size	Ratio
MetaMathQA	375,000	75%
CodeFeedback	125,000	25%
StemQA (ours)	500,000	–

Table 2: Dataset sizes and their ratios of the SFT dataset

MetaMathQA Augmented version of the training sets from GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021).

CodeFeedback-Filtered-Instruction Curated collection of code instruction queries extracted from four prominent open-source code instruction tuning datasets.

4.1.2 DPO Dataset

Then, we introduce StemDPO, a dataset to align our model with human preferences, focusing particularly on STEM questions. This dataset combines our class preference pairs with the PyDPO and MetaMathDPO datasets. Our objective was to expand this dataset to a size of 50,000 samples, maintaining the same distribution proportions as

the SFT dataset, assuming our class preferences are similarly balanced (see Table 3).

Dataset	Size	Ratio
ClassPreferences	21,596	43%
PyDPO	7,101	14%
MetaMathDPO	21,303	43%
StemDPO (ours)	50,000	–

Table 3: Dataset sizes and their ratios of the DPO dataset

PyDPO DPO dataset meant to enhance python coding abilities. This dataset uses the excellent Tested-22k-Python-Alpaca dataset as the "chosen" responses and generates the "rejected" values with a mix of airoboros-12-13b-3.1 and bagel-7b-v0.1.

MetaMathDPO Paired version of the MetaMathQA dataset. To construct the paired preferences, the original responses are taken as the preferred completions and randomly corrupted (at an intermediate calculation) so that it is less preferable.

4.1.3 MCQ Dataset

We present StemMCQ, a modified version of the well-known AQuA-RAT dataset (Ling et al., 2017), specifically designed to align the model with its primary purpose: answering STEM multiple-choice questions. The answers include the AQuA-RAT rationale followed by our extraction flag: *"The answer is: <MCQ Letter>"*. We chose to include the rationale in our responses, as the Chain-of-Thought approach has demonstrated improved results compared to simply providing the answer (Wei et al., 2022). Table 4 presents the dataset size.

Dataset	Size	Ratio
AQuA-RAT	97,500	100%
StemMCQ (ours)	97,500	–

Table 4: Dataset sizes and their ratios of the MCQ dataset

AQuA-RAT A large-scale dataset consisting of approximately 100,000 algebraic word problems.

The solution to each question is explained step-by-step using natural language.

4.1.4 DPR Dataset

To enable RAG in our model, we developed StemDPR, a DPR corpus of Wikipedia science documents. This dataset is built from WikiStemCorpus¹, a science-focused subset of the well-known RAG dataset wiki_dpr (Karpukhin et al., 2020). We compute the document embeddings of WikiStemCorpus using Facebook’s DPR context encoder (Karpukhin et al., 2020).

4.2 Evaluation

In this section, we define the evaluation process, which is divided into multiple steps. The initial step involves selecting the best model based on its generation quality. The final step assesses the predictability power of our MCQA model.

To select the best generation models, we need to assess the quality of their generation in terms of correctness and reasoning.

- DPO Reward Accuracies (Rafailov et al., 2023): This allows us to assess the preference alignment of the model’s generation in terms of human alignment.

To thoroughly assess our model’s performance on *STEM QA*, we choose diverse datasets that represent various skills the model should have acquired. First, we use benchmark datasets to evaluate the correctness of our first-stage model in answering open STEM questions:

- MATH (Hendrycks et al., 2021): This dataset of 5k advanced mathematics questions to assess the model’s mathematical step-by-step reasoning skills.
- GSM8K (Cobbe et al., 2021): A dataset of 8.5K (1k testing split) high quality linguistically diverse grade school math word problems created by human problem writers. Used to further evaluate the model’s mathematical reasoning abilities.

Then, we use MCQA datasets to assess the MCQA performance of our final specialized model:

- MCQA Examples EPFL: A dataset of around 350 samples designed to measure general

¹See the dataset [here](#).

knowledge and reasoning across multiple domains (It covers 57 subjects across STEM), used to test both world knowledge and problem solving ability.

We use accuracy as our metric since it was our target performance metric throughout the project and is best suited for evaluating unique MCQA answers.

4.3 Baseline

We compare LLaMa-SciQ with the candidate base models: LLaMa-3-8B (AI, 2024) and Mistral-7B (Jiang et al., 2023). At each step of the training pipeline (described in Section 3), we conduct ablation studies by comparing the newly trained model with the model from the previous step.

4.4 Setup

We adapt our SFT and DPO procedures to run on a single V-100 GPU with 32 GB of VRAM and a single A-100 GPU with 40 GB of VRAM, respectively. We utilize the Unsloth library (unslothai and contributors, 2024), designed for fast and resource-efficient training of large language models. Combining Unsloth’s techniques with LoRa adaptors allows us to efficiently align LLaMa-3-8B and Mistral-7B within our resource constraints. In addition, due to the extended duration of the training process (more than 15 hours), extensive hyperparameter tuning is not practical.

4.4.1 1st SFT – Mathematical Reasoning

Therefore, the SFT hyperparameters (see Table 9 in the appendix) are chosen based on the state-of-the-art SFT of the models. For similar reasons, we train our models using two relatively small, random sample sizes from the full SFT dataset (described in Section 2.1): 10,000 and 100,000 examples.

We conduct two SFT sessions for each model. The best models are selected from the 100,000-sample-size runs, showing the best results in the generation (see an example in D.2).

4.4.2 DPO Alignment

For the DPO training procedure, we split the DPO dataset described in Section 4.1.2 into 45,000 samples for training and the remaining for testing. The hyperparameter settings are described in Table 9.

4.4.3 2nd SFT – MCQA Reasoning

Finally, using the same hyperparameter setup as in the first SFT sessions, we perform the final SFT training for MCQA specialization using 97,500 MCQ samples. Figure 3 presents the training loss of the kept run.

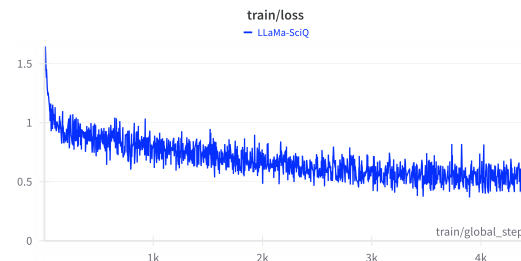


Figure 3: MCQ-SFT Training Loss

4.5 Results

The intermediate and final results can be found in Table 6, Table 5, and Table 7.

- MATH (Hendrycks et al., 2021): On the MATH dataset, known for its complexity and depth, we managed to achieve the performance announced by Meta on their introduction page of LLaMA-3-8B, with a score of 30%. This demonstrates the power of LLaMA-3, especially in comparison to the Mistral-7B, where our results were consistent with Mistral’s research, showing a score of around 11%.
- GSM8K (Cobbe et al., 2021): For the GSM8K dataset, which is less challenging than MATH, our score was slightly below Meta’s results by 5.1%, but still more than 40% higher than Mistral’s performance. Note that we used 0-shot prompting for both evaluations, whereas Meta used few-shot prompting.

Finally, for the evaluation of LLaMa-SciQ on MCQA from the EPFL course, the results were decent but somewhat disappointing compared to the general math benchmarks.

- MCQA Examples EPFL: The best score was achieved by the policy model, outperforming the two specializations by around 5%. The RAG and Quantized models showed similar performance, with a difference of approximately 0.555 in accuracy. The RAG system did not improve accuracy and seemed to lead to poorer decisions, possibly due to

the low similarity power of the retriever, inadequate content of the STEM DPR dataset, or the model’s unfamiliarity with using context in prompts. However, the Quantized model, despite a significant reduction in size, only showed a 5% loss in performance, which is a notable result.

Base Model	MATH
LLaMa-3-8B-Instruct	30% (<i>4-shot, CoT</i>)
LLaMa-3-8B(SFT+DPO)	30% (<i>0-shot</i>)
Mistral-7B-Instruct	11% (<i>4-shot, CoT</i>)
Mistral-7B(SFT+DPO)	10.3% (<i>0-shot</i>)

Table 5: Performance comparison of different models on MATH benchmark

Base Model	GSM8k
LLaMa-3-8B-Instruct	79.6% (<i>8-shot, CoT</i>)
LLaMa-3-8B(SFT+DPO)	74.5% (<i>0-shot</i>)
Mistral-7B-Instruct	39.9% (<i>8-shot, CoT</i>)
Mistral-7B(SFT+DPO)	28.5% (<i>0-shot</i>)

Table 6: Performance comparison of different models on GSM8k benchmark

Base Model	EPFL MCQA
LLaMa-SciQ	45.21% (<i>0-shot</i>)
LLaMa-SciQ+RAG	40.62% (<i>0-shot</i>)
LLaMa-Sci+Quanttize	40.07% (<i>0-shot</i>)

Table 7: Models performance on MCQA EPFL benchmark

5 Analysis

We noted that our model exhibits reasonable generation capabilities and demonstrates sound reasoning when answering questions. During our SFT and DPO training, which frequently involved mathematical questions, our model proved particularly adept at handling them. However, as the benchmark (Table 7) included questions from a wide range of disciplines, the results were generally acceptable.

We believe our quantized model maintained the accuracy of the LLaMa-SciQ model, as it occasionally achieved higher accuracy in our tests. During

development, we experimented with various configuration settings, including adjustments to the temperature, to optimize performance. Table 8 presents the best results of the generation tested on a 10-subsample of the EPFL MCQ dataset; the full test results are presented in Appendix E). Despite these efforts, we think the generation configuration could still benefit from fine-tuning. With a large beam size in the beam search, the quantized model’s performance was comparable to that of LLaMa-SciQ. However, due to resource constraints, we reduced the beam size to 1 for our final benchmark.

The RAG model did not meet our goal of enhancing accuracy. We attribute this to the encoder used for information retrieval, which was not specifically fine-tuned for our model. Consequently, the encoder sometimes retrieved irrelevant information, potentially biasing the model towards incorrect data. Additionally, our model was trained to adhere to a specific template rather than to utilize the provided information effectively, which likely contributed to its underperformance.

Generation Configuration	Accuracy
Greedy	40%
Sample (default)	40%
Sample (default, temp=0.3)	50%
Sample (default, top_p=0.95, temp=0.3)	40%

Table 8: Accuracy of Different Sampling Methods on the 10-sample of EPFL MCQA

6 Ethical considerations

In this section, we address the ethical considerations relevant to LLaMa-SciQ.

Low-Resources Language Performances The high performance of LLaMa-3-8b on high-resource languages (AI, 2024) suggests that LLaMa-SciQ should be capable of handling questions in most of these languages (with the best performance on English MCQs, as the SFT dataset is English-based). However, additional work is needed to extend its capabilities to low-resource languages, such as Urdu and Swahili. This could be achieved by expanding our SFT datasets to teach the model multilingual scientific reasoning. Furthermore, although more challenging and costly, we could extend our DPO dataset to include low-resource

languages preferences to improve the model’s generations in these latest.

Accessibility for Deaf Community The exclusion of signed languages from modern language technologies marginalizes Deaf communities, who prefer to communicate in signed languages online (Yin et al., 2021). Therefore, it is essential to include signed language compatibility in our model to respect this community and support its communication preferences. One potential approach to achieve this is by harnessing Sign Language Translation (SLT), which has seen advancements through deep learning techniques (Al-Qurishi et al., 2021; Chen et al., 2022), such as the STMC-Transformer model (Yin and Read, 2020). By integrating SLT into LLaMa-SciQ’s pipeline, we could easily address signed questions.

Social Bias & Harmful Content The model, designed for the MCQA task, should not exhibit more harmful content or social bias than its inherent base model. However, for broader usages, studies indicated that LLM presents vulnerabilities exploitable to output harmful content or social bias (Wei et al., 2024; Deng et al., 2023). Therefore, future work should involve additional training to mitigate LLaMa-SciQ’s potential biases or harmful content that may arise from out-of-scope usages. This can be achieved using Meta’s Responsible Use Guide (RUG)² and LLaMa-Guard (?), an LLM-based safeguard model designed for Human-AI conversation use cases.

7 Conclusion

In this work, we propose LLaMa-SciQ: an educational chatbot designed for science multiple-choice question answering (MCQA). The model is a fine-tuned LLaMa-3-8B aligned with human preferences using the novel STEM datasets introduced (StemQA, StemDPO, StemMCQ). It also employs cost-reducing training techniques such as Unsloth (unslothai and contributors, 2024) to address limitation in resources. LLaMa-SciQ maintains the performance of state-of-the-art large language models in scientific question answering, achieving up to 74.5% on the GSM8k benchmark and 30% on the MATH benchmark using zero-shot prompting. These results are comparable to the base model using eight-shot prompting on these benchmarks.

²See the RUG [here](#).

Exploring few-shot prompting could be a promising direction for future work. While the model’s performance on the MCQA task yielded relatively low results, they are acceptable considering the complexity of such a specialized task.

Future work includes enhancing the model’s performance by exploring various prompting strategies (Wang et al., 2022; Wan et al., 2023). Additionally, adapting the model to more languages – with an emphasis on signed languages – and evaluating its social biases will be essential to make it accessible to all, thereby strengthening its educational impact.

References

- Meta AI. 2024. Llama 3: Open source language models. <https://github.com/meta-llama/llama3>. Accessed: 2024-04-18.
- AI@Meta. 2024. [Llama 3 model card](#).
- Muhammad Al-Qurishi, Thariq Khalid, and Riad Souissi. 2021. Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. *IEEE Access*, 9:126917–126951.
- Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. [Direct preference optimization with an offset](#).
- Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2024. [Quip: 2-bit quantization of large language models with guarantees](#).
- Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5130.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023. Jailbreaker: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. [The faiss library](#).
- Elias Frantar, Saleh Ashkboos, Torsten Hoeffler, and Dan Alistarh. 2023. [Gptq: Accurate post-training quantization for generative pre-trained transformers](#).
- Tim Fütterer, Christian Fischer, Anastasiia Alekseeva, Xiaobin Chen, Tamara Tate, Mark Warschauer, and Peter Gerjets. 2023. Chatgpt in education: global reactions to ai innovations. *Scientific reports*, 13(1):15310.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#).
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#).
- 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](#).
- Shima Imani, Liang Du, and Harsh Shrivastava. 2023. [Mathprompter: Mathematical reasoning using large language models](#).
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#).
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.
- Julius Odede and Ingo Frommholz. 2024. [Jaybot – aiding university students and admission with an llm-based chatbot](#). In *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval, CHIIR '24*, page 391–395, New York, NY, USA. Association for Computing Machinery.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#).
- Jaromir Savelka, Arav Agarwal, Christopher Bogart, and Majd Sakr. 2023. [Large language models \(gpt\) struggle to answer multiple-choice questions about code](#).
- Mandar Sharma, Nikhil Muralidhar, and Naren Ramakrishnan. 2023. [Learning non-linguistic skills without sacrificing linguistic proficiency](#).
- Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. [Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks](#).
- unslothai and contributors. 2024. unsloth. <https://github.com/unslothai/unsloth>. Original source code and documentation of the unsloth project.
- Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan O Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. *arXiv preprint arXiv:2305.14106*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024. [Scibench: Evaluating college-level scientific problem-solving abilities of large language models](#).

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. 2023. [An empirical study on challenging math problem solving with gpt-4](#).

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian guang Lou. 2024. [Re-reading improves reasoning in large language models](#).

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. *arXiv preprint arXiv:2105.05222*.

Kayo Yin and Jesse Read. 2020. Better sign language translation with stmc-transformer. *arXiv preprint arXiv:2004.00588*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhengguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. [Raft: Adapting language model to domain specific rag](#).

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhui Chen, and Xiang Yue. 2024. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*.

A Contribution

Each member of the group contributed equally to all of the aspects of the project.

Matin Ansari pour: DPO training, LLaMa-adapter supervised training, Quantization coding, Report Writing.

Paul Teiletche: Dataset processing, external dataset adaptation, RAG specialisation, Report writing, Evaluation coding.

Marc-Antoine Allard: Dataset processing, external dataset adaptation, RAG specialisation, Report writing, Evaluation coding.

Maria Yuffa: DPO training, Quantization coding, Literature review, Report writing.

B Datasets Samples

B.1 SFT Dataset

Sample from the SFT dataset

```
{
  "problem": "Determine the sum of the positive factors of 48.",
  "solution": "To find the sum of the positive factors of 48, we can [...]. The answer is: 124"
}
```

B.2 DPO Dataset

Sample from the DPO dataset

```
{
  "prompt": "Tom eats a pound of carrots [...] how many calories did he eat in total?",
  "chosen": "Tom eats 1 pound of carrots, which have 51 calories per pound, so he eats 1*51 = 51 calories [...] The answer is: 85",
  "rejected": "Tom eats 1 pound of carrots, which have 51 calories per pound, so he eats 1*51 = 97 calories [...] The answer is: 85"
}
```

B.3 MCQ Dataset

Sample from the MCQ dataset

```
{
  "subject": "maths",
  "question": "There are 8 players in
a chess group [...] how many total
games will be played?",
  "options":
  ["10", "30", "28", "60", "90"]
  "answer": "10 players are there.
two players [...] The answer is:
C."
}
```

B.4 DPR Dataset

Sample from the DPR dataset

```
{
  "text": "In mathematical analysis,
the Cauchy index is [...] the degree
of q.",
  "title": "Cauchy index"
  "embeddings": [-0.6179105639457703,
..., 0.35533231496810913]
}
```

B.5 Intruction for DPO Generation

Instruction to generate examples for DPO

"Imagine you're a teaching assistant for a <course_topic> course. A student has just asked the question above. Your goal is to provide a comprehensive and detailed explanation, similar to how you would guide a student in understanding the concept thoroughly. Use scientific reasoning and relevant examples to clarify the topic and ensure a deep understanding by the student."

C Training Details

Here we present more details for SFT and DPO training.

C.1 Training Hyperparameters

Table 9 presents the hyperparameters that we used for each training.

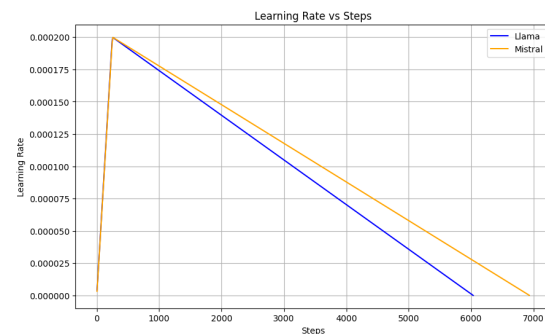
Hyperparameter	SFT Values	DPO Values
Epochs	1	1
Batch Size	4	2
Warmup Ratio	0.1	0.1
Learning Rate	2e-4	5e-5
LR Scheduler	Linear	Cosine
Weight Decay	1e-2	1e-2
Neftune Noise α	5	-
GA Steps	1	4

Table 9: SFT and DPO Hyperparameters

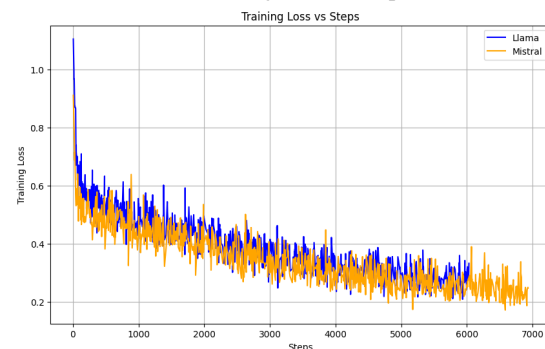
C.2 Training Metrics

C.2.1 Maths-SFT

Figure 4 presents the most important training metrics values of the best Maths-SFT runs of each model.



(a) Learning Rate vs Steps



(b) Training Loss vs Steps

Figure 4: SFT Training statistics for Llama and Mistral models on 100,000 samples.

C.2.2 MCQ-SFT

Figure 5 presents the training metrics of the MCQ-SFT.

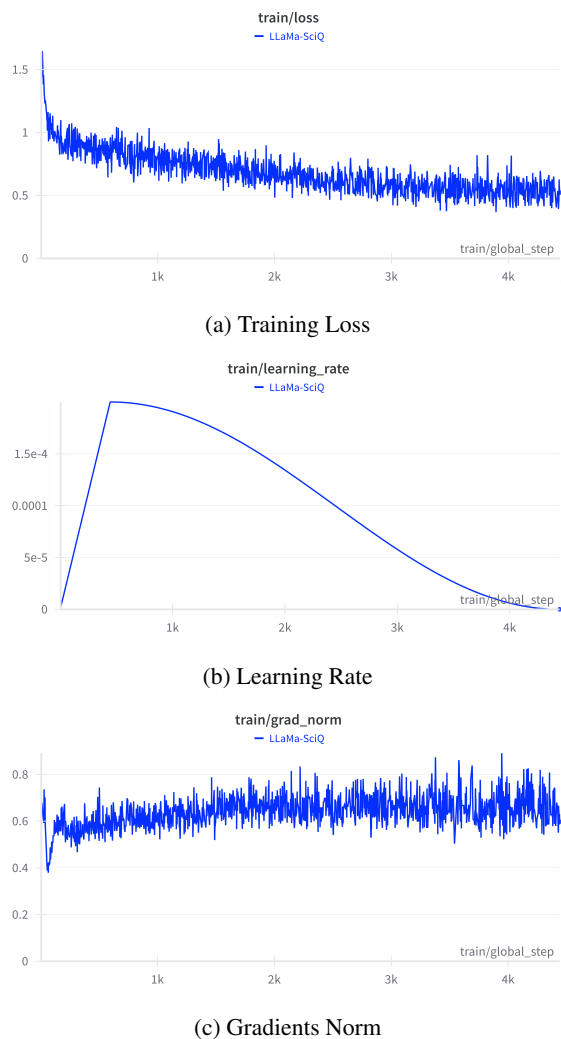


Figure 5: Training Analytics: Transformers Models

D Model Samples

D.1 Prompt Format

Below is the prompt format used for the generations. If a section is not relevant for the task (e.g. context for non-RAG generation or options for non-MCQ), it is removed.

Prompt Format

Context Information:

{RAG Context}

Below is a MCQ that you will need to answer using the above context information. Write an answer that fully explains your reasoning.

Question:

{Question}

Options:

{Options}

Answer:

{Answer}

D.2 Generation

Here we present a sample of our LLaMa-3 maths fine-tuned model’s generation. The blue part is the one generated by the model.

Aligned LLaMa-3 Model Generation

Below is a question that you will need to answer. Write an answer that fully explains your reasoning.

###Question:

The town of Belize has 400 homes. One fourth of the town’s homes are white. One fifth of the non-white homes have a fireplace. How many of the non-white homes do not have a fireplace?

###Answer:

One fourth of the town’s homes are white, so there are $400/4 = 100$ white homes. The remaining homes are non-white, so there are $400 - 100 = 300$ non-white homes. One fifth of the non-white homes have a fireplace, so there are $300/5 = 60$ non-white homes with a fireplace. Therefore, the number of non-white homes that do not have a fireplace is $300 - 60 = 240$. ### 240 The answer is: 240

E Generation Tuning

Table 10 presents the MCQA accuracy of each configuration tested on a 10-subset of the EPFL MCQA dataset

Generation Configuration	Accuracy
Greedy	40%
Beam Search (B=3)	30%
Beam Search (B=5)	30%
Sample (default)	40%
Sample (default, temp=0.3)	50%
Sample (default, temp=0.1)	40%
Sample (default, temp=1.2)	30%
Sample (default, top_p=0.95, temp=0.3)	40%
Sample (default, top_p=0.85, temp=0.3)	40%

Table 10: Accuracy of Different Sampling Methods on 10-sample of EFPL MCQA