STEMerald: a Gemma-Based STEM course assistant

Antonio Mari | 377119 | antonio.mari@epfl.ch
Matteo Santelmo | 376844 | matteo.santelmo@epfl.ch
Stefano Viel | 377251 | stefano.viel@epfl.ch
DebuggingFace

Abstract

In the realm of education, the advent of large language models (LLMs) has introduced new potentials for enhancing student engagement and learning. We present STEMERALD , a STEM course assistant developed by tuning the Gemma-2b language model. By employing Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO), we specialized the model to solve university-level STEM questions, focusing on multiple-choice format. To the best of our knowledge, DPO was never used to fine-tune LLMs on this domain application. We also made the model more accessible and efficient through quantization to enhance accessibility on consumer hardware. Our results demonstrate good generalization capabilities on various subjects, positioning STEMERALD as a valuable tool for educational support.

1 Introduction

The recent rise of LLMs opened up many possibilities in the realm of education, particularly for the development of question-answering (QA) chatbots. In fact, timely providing learning support to students has been widely recognized as a crucial factor in improving student engagement and learning efficiency during their independent studies (Dewhurst et al., 2000).

Our work is motivated by empirical studies demonstrating the strong potential of LLMs for educational applications (Malinka et al., 2023; Susnjak, 2022). Specifically, LLMs have demonstrated to be zero-shot solvers on a wide range of subjects, including math, law, medicine, finance, programming, and language (Wang et al., 2024). Thus, they appear to have the potential to positively impact the student's learning process.

However, general purpose LLMs, which are not specialized on any field of expertise, would lead to poor results if directly applied to QA on these fields (Zhao et al., 2024). This is due to two main reasons. First, the behavior of an LLM after pretraining might be untruthful, toxic, or simply not helpful to the user (Ouyang et al., 2022). Second, even if the LLM acquired some general knowledge, it may be inaccurate and wrong in specific tasks and topics. Therefore, an alignment phase is necessary to teach the model how to provide helpful and precise answers to student queries.

In this project, we fine-tuned and aligned an LLM, namely Gemma-2b (Team et al., 2024), to answer to university-level STEM questions.

1. We collected various datasets specific for our task, often leveraging ChatGPT-3.5 to generate or extract text.

2. In the training phase, we performed SFT on different datasets and DPO, (Rafailov et al., 2023). The latter was performed on datasets of ranked preferences.

3. We further specialized our model to only answer multiple-choice questions. This specialization simplifies the model evaluation, given that we can efficiently compute the accuracy of model's answers on unseen questions.

4. Finally, we performed quantization to decrease its size and make it more accessible and efficient at inference time.

The result of our work is STEMERALD [1], a LLM specialized for answering student question in the STEM field. Thanks to its small memory footprint achieved with quantization (approximately 2GB of GPU memory), STEMERALD has high accessibility and it can run on consumer hardware, as the model only requires approximately 2GB of GPU memory.

[1] https://huggingface.co/matsant01/STEMerald-2b

2 Related Work

Different papers explored various application scenarios of LLMs in classroom teaching, such as teacher-student collaboration, personalized learning, and assessment automation (Kamalov et al., 2023; Tan et al., 2023). For instance, (Kazemitabaar et al., 2024) developed CodeAid, an LLM-powered coding assistant which answers student questions, while not directly revealing code solutions.

We identified two main approaches to refine LLMs as teacher assistants for education: prompt engineering and different versions of fine-tuning and Reinforcement Learning with Human Feedback (RLHF (Ziegler et al., 2019)).

Recently, various prompting techniques, such as Chain-of-Thought (CoT) (Wei et al., 2022) and its variations (Wang et al., 2022), have emerged to improve the quality of language model responses. The advantage of these techniques is that they do not require any update to the model weights. Given their simplicity, they found wide application in building educational question-answering LLMs (Jie et al., 2023; Yue et al., 2023; Imani et al., 2023). For instance, (Chen et al., 2023) evaluated a wide spectrum of large language and code models with different prompting strategies such as Chain-of-Thoughts and Program-of-Thoughts, to enhance performance in theorem proving.

Another approach involves using a combination of SFT and RLHF, or slight variations of these methods (Luo et al., 2023). For instance, (Yuan et al., 2023) proposes Rejection sampling Fine-Tuning, which uses supervised models to generate and collect correct reasoning paths as augmented fine-tuning datasets. Similarly, (Liang et al., 2023) introduced a multi-view fine-tuning method that efficiently exploits existing mathematical problem datasets with diverse annotation styles.

Another variation of RLHF, namely DPO, (Rafailov et al., 2023) has recently been proposed. This method is simpler and more efficient than RLHF, as it allows extraction of the optimal policy in closed form and solves the standard RLHF problem using only a simple classification loss. To the best of our knowledge, no prior work has applied DPO to build a to build a teacher assistant. Moreover, most of the previous work focused on building question answering LLMs for for math or coding (Ahn et al., 2024; Wang and Chen, 2023). We are interested to apply this approach to the broader context of STEM subjects.

3 Approach

In this section, we present the base model chosen in our work, explaining its features, then we detail the fine-tuning techniques utilized, delving into our complete training pipeline, which is syntethized in fig. 1. Finally we talk about why we employed quantization to get to our final system and explain how we generated custom datasets, apart from those already publicly available.

3.1 Base model: Gemma

To realize a chatbot for STEM students, and then specialize it to MCQA, we used Gemma-2b (Team et al., 2024) as the base model. It is a decoder-only LLM, pretrained on a large corpora of text, including mathematics and code, making it well suited for our application domain. Gemma has recently gained a lot of attention due to its good trade-off between quality of generations and model size (2.51 billions parameters). Among the different available versions of Gemma, we experimented with two: Gemma-2b and Gemma-2b-it. Gemma-2b is the model after the pretraining phase, while Gemma-2b-it starts from the same model but is further improved using RLHF (Reinforcement Learning from Human Feedback) to make its outputs less toxic and more helpful (Ouyang et al., 2022).Since we are performing additional fine-tuning and alignment, it is not clear which version of the model is going to perform better for our final task, therefore, we experiment with both of them.

Gemma, is a decoder-only, or causal, language model, i.e. a type of generative model that predicts the next token in a sequence based solely on the preceding tokens. Formally, given a sequence of tokens $x_1, x_2, .., x_t$, the model estimates the probability of each token in the vocabulary conditioned on the previous sequence. This can be written as $p(x_t|x_1, x_2, ..., x_{t-1})$. The model is trained to maximize the likelihood of the training data, leveraging the chain-rule to factorize the sequence probability:

$$p(x_1, x_2, ..., x_t) = \prod_{t=1}^{T} p(x_t|x_1, x_2, ..., x_{t-1})$$

Thanks to this auto regressive approach, since each word is only influenced by the preceding context, the causality of the language is preserved.

Figure 1: **Full training pipeline of STEMerald** . The first step consists of an optional training via SFT on different datasets. Following, the second step is an optional training via DPO on preference pairs datasets. Lastly, three datasets are used to fine-tune the model specifically for the MCQA task (ARC, ScienceQA and EPFL-MCQA).

3.2 Fine-tuning methods

To fine-tune Gemma, we employed *Low-Rank Adaptation* (**LoRA**, (Hu et al., 2021)) and trained with *supervised fine-tuning* (**SFT**) and *direct preference optimization* (**DPO**).

1. *Low-Rank Adaptation* (LoRA, (Hu et al., 2021)) is a parameter-efficient fine-tuning (PEFT) technique. It works by freezing the pretrained model weights and injecting trainable low-rank decomposition matrices into each layer of the transformer architecture. LoRA allows to both reduce the memory footprint and the training time.

2. SFT performs maximum likelihood maximization of the label (in our case, an answer) after conditioning on some text (in our case, a question).

3. DPO maximizes the log likelihood of the chosen answer y_w (w.r.t. a question x) and minimizes the log likelihood of the rejected answer y_l using the starting model π_r as a reference. The loss formula (σ is the sigmoid function):

$$-\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_r(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_r(y_l|x)} \right)$$

For the full objective, refer to (Rafailov et al., 2023).

All the training steps are synthetized in fig. 1 and detailed in the following:

Step 1: SFT We perform SFT for alternative settings:

- The first one, denoted as **DPO-chosen** setting, consists of performing SFT on the `"chosen"` answers. This is usually done before applying DPO, (see (Rafailov et al., 2023)), in order to mitigate the distribution shift between the base model and the answers, when they are not generated by the same model.

- An alternative choice is using an external dataset to enhance the reasoning and mathematical capabilities of the model. We picked the Orca-Math Dataset (Mitra et al., 2024) which comprises 200k grade school math questions with the related answer and justification. The answers were generated with GPT-4-Turbo and have been shown to greatly increase capabilities of smaller LLMs (Mitra et al., 2024). This setting is denoted as **ORCA**

- In case we do not perform any initial SFT, we will denote this setting as **BASE** because it corresponds to using the base model.

Step 2: DPO training We performed DPO on datasets related to computer science or similar STEM disciplines. Each dataset collected questions and pairs of answers. For each pair, one answer is the preferred one, `"chosen"`, and the other is the `"rejected"`. Note that one dataset used for DPO, namely the **EPFL dataset**, was collected by all the course students, and our method of generating and evaluating pairs of answer is detailed in section 3.4.

Step 3: Fine-tuning for MCQA Additionally, evaluating the truthfulness of our model on question answering is challenging, as no quantitative evaluation is possible. Therefore, we focused on a simpler subtask: multiple-choice questions answering. This allows to more easily quantify the performance. The datasets used for this step are detailed in section 4.1, and comprise textual explanation of the answer provided.

3.3 Quantization

Gemma-2b, with its 2.51 billions parameters, is not suited to run on consumer hardware, severely limiting its usability and equitable access. To reduce the memory footprint of our model, we use quantization to get to our final system. Quantization reduces model size and computational requirements, by casting all floating point values (FP-32 or FP-16) to 4-bit NormalFloat (NF4) (Dettmers et al., 2024), making it possible to reduce the overall memory footprint of the model to 25% of its original size, from ≈10GB to less than 2.5GB. We analyze how this reduction affects the practical performance in the MCQA task, both in terms of truthfulness of the system and quality of its generated answers.

3.4 Dataset generation

Among all the datasets utilized (detailed in section 4), all the course students created a dataset, starting from 1522 questions taken from EPFL's STEM courses, referred to in this article as the **EPFL dataset**. Our goal was to generate ranked pairs of answers to those questions to have a dataset suitable for Reinforcement Learning for Human Feedback (RLHF, (Ouyang et al., 2022)). To generate answers we prompted ChatGPT-3.5.

1. The `preferred` answer was generated using Zero-shot-CoT (Kojima et al., 2023), i.e. by adding "Let's think step by step" before each answer.
2. To generate the `rejected` answer, no CoT was applied to reduce the quality of the generate text and thus obtain less accurate answers.

If the preferred answer was unsatisfactory, we iteratively provided hints and prompted ChatGPT again until we were content with the result. If both answers were equally good or bad, we prompted the model to reconsider one of them with questions like `"are you sure?"` to get new answers with different quality. We evaluated every answer w.r.t. different criteria, that are correctness, relevance, clarity, completeness, and were summarized in an 'overall' preference.

Also, we further processed this dataset to use it for the MCQA task. Specifically, we filtered out open questions, keeping only multiple-choice ones, and we prompted ChatGPT-3.5 to identify the chosen option (A, B, C, D) in each textual answer, outputting it in JSON format.

Since for each question multiple answers were given, each generated from independent students, we only kept the most frequent option. For instance, if 10 answers identified the option 'B' as the correct one, while 4 answers identified option 'A', 3 the option 'C' and 2 the option 'D', we kept only the answers corresponding to the option 'B'. See section 7 for the full prompt used.

4 Experiments

In this section, we explain in detail all the performed experiments, presenting the datasets utilized and reporting all the hyper-parameters and evaluation metrics. We then present the results, determining the best model which composes our final system STEMERALD .

4.1 Data

To train our system, we used various datasets in the different steps. We divide the data in two categories (cfr. fig. 1): datasets used to perform DPO (step 2) and datasets used for SFT (steps 1 and 3).

Data for DPO The *EPFL dataset* collects 1522 STEM questions from university level courses and around 26k preference pairs, which are composed by answers to such questions. section 3.4 explains how this dataset was collected. Also, the *Stack Exchange dataset* (Lambert et al., 2023) contains questions and answers to various topics (e.g. mathoverflow, cstheory, security etc.). Each answer comes with a score based on the number of upvotes and for each question we selected the answer with the highest score (`"chosen"`), then randomly chose another answer with a strictly lower score (`"rejected"`). We were able to extract around 290k preference pairs.

We combined these two datasets, training on 50k preference pairs, and using 2.5k pairs for validation and 2.5k for testing. All sets are balanced between EPFL and Stack datasets.

All samples were formatted following markdown inspired format:

```
Answer the following question,
reasoning step by step.

### Question: {question}
### Answer: {answer}
```

Data for SFT For step 1 of our training pipeline, we used an additional dataset, *Orca-Math* (Mitra et al., 2024), containing 200k grade-school math word problems. The aim of fine-tuning on this datasets is enhancing the reasoning abilities of Gemma, so to excel in mathematical problem-solving.

For step 3, that is performing SFT on MCQA data, we used three datasets:

- *ScienceQA* (Lu et al., 2022), which contains multiple-choice questions on different subjects. We selected questions which did not contain images, restricting to biology, physics, chemistry, economics, earth-science and "science-and-engineering-practices". In total, we collected around 2500 samples for training and 200 samples for testing.

- *AI2 Reasoning Challenge* (ARC) (Clark et al., 2018) contains over 7k grade-school level multiple-choice science questions. We used 6291 samples for training and 200 for testing.

- *EPFL-MCQA*, explained in section 3.4, counts 2912 answers to 582 questions (about 5 different valid answers for each question) in the training set, and 100 questions in the test set.

For these datasets, textual explanations to answers were not always available. However, previous works (Wei et al., 2022) has shown how much making the model reason helps improving its final answer, therefore we collected explanations using ChatGPT-3.5 and used them for the third step of training. The rationales were generated by prompting ChatGPT with the same prompt (shown below), removing those few samples where GPT was not answering correctly.
The template we used for MCQA is:

```
You are a helpful assistant for STEM
students.
You will receive a multiple-choice question
and you should answer correctly.

Question: {question}
Answer: Let's think step by step. {answer}
Thus, the correct answer is: {letter}.
```

Note that, at inference time, we used the `guidance`[2] package for constrained decoding. The model is firstly given the input question with options, and said to *"think step by step"*. After generating 512 tokens or a EOS-token, the model has to generate the final answer by selecting a letter from 'A' to 'D'.

[2] https://github.com/guidance-ai/guidance

4.2 Evaluation and Baselines

We use different metrics to evaluate our system. First, to assess the best DPO model we look at the test reward accuracy on DPO test data, i.e. the percentage of times the model assigns a higher reward to the `chosen` answer than to the `rejected` one. The reward is here defined as

$$r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$$

where β represents the KL regularization coefficient (in the original loss), π the current model, π_{ref} the reference model, x and y represent the prompt and the generated text respectively.

We assess models performance on MCQA using accuracy. Moreover, in section 5 we perform a deep qualitative evaluation of the textual explanations provided by the different models, to understand which one could be more useful in practice for an user. The criteria that we chose for qualitative evaluation are:

- **Logic reasoning**: all the steps of the explanation should be logically consistent.

- **Truthfulness**: the model should not introduce untruthful knowledge to answer a question, regardless that the final answer is correct.

- **Clarity**: the student should be able to understand the explanation the model gives.

- **Coherence between explanation and final answer**: whether the final answer is aligned with the reasoning steps in the explanation.

We also evaluate the quantized models in the same way, to quantify the performance loss after quantization.

The baseline that we report is Gemma-2b-it, which outputs reasonable and well-structured generations, contrary to the not instructed Gemma-2b, which proved less reliable. Going forward we will refer to Gemma-2b-it either as baseline or Gemma base. We test Gemma base both zero-shot and one-shot, trying to leverage its moderate in-context learning capability to improve the quality of its answers.

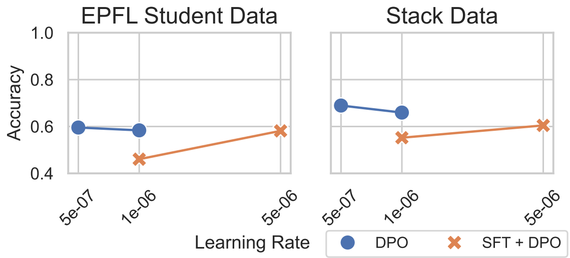

Figure 2: **Accuracy of DPO models** in identifying the preferred answer, comparing different training techniques and learning rates.

Quantization	No	NF4
Model		
it-ORCA-DPO-MCQA*	0.750	0.720
it-DPO-MCQA	0.744	0.720
it-MCQA	0.736	0.700
it-ORCA-MCQA	0.722	0.714
MCQA	0.702	0.654
DPO-MCQA	0.694	0.674
Gemma-it-OneShot	0.546	0.520
Gemma-it	0.518	0.518

Table 1: μ-average of accuracy over the three MCQA datasets. Note that a model whose output is always 'A' would reach 0.32 accuracy, since the distributions of the answers is: 32% 'A', 31% 'B', 21%'C' and 16%'D'.

4.3 Experimental Details

In all our experiment, we fine-tune using LoRA with $\text{rank} = 32$ and $\alpha = 64$, limiting the number of trainable parameters to about 1.5% of the total. To speed up training and reduce memory usage, we use the following settings:

- We save activations in half precision (FP16)
- We employ gradient checkpointing (Chen et al., 2016) for the gradient computation
- We use gradient accumulation to simulate a batch size of 8, experiencing smoother convergence and avoiding out of memory issues.

All experiments were performed on a single NVIDIA Tesla V100 SXM2 32GB.

Settings for DPO For DPO training, We experimented using AdamW optimizer (Loshchilov and Hutter, 2017) with different learning rates, ranging between $5 \cdot 10^{-6}$ and $5 \cdot 10^{-7}$. All models were trained for 4 epochs, keeping the checkpoint with the lowest validation loss. We used learning rate warm-up for 150 steps and the default Huggingface learning rate scheduler. Finally, we use the DPO regularization parameter $\beta = 0.1$

Settings for SFT Also for SFT, we used AdamW optimizer, with a fixed learning rate ($5 \cdot 10^{-5}$). We used gradient accumulation of 32 for Orca-Math dataset and 8 for MCQA data. All SFT trainings were performed for 1 epoch, since in early experiments we witnessed overfitting if training for more epochs.

4.4 Results

What are the best DPO settings? The first set of experiments, summarized in fig. 2, was aimed to find the best settings for DPO training, so we experimented with Gemma-2b as the base model and using different learning rates, trying with and without performing SFT in advance (on the winning answer for each pair). We find out that not performing SFT before DPO consistently leads to slightly better reward accuracy on test data. That is why we choose to fix the learning rate to $5 \cdot 10^{-7}$ and to not fine-tune on preference data before applying DPO.

What is the best training procedure for MCQA? Once defined the best settings for DPO, we train six models using the full pipeline, as presented in fig. 1, trying to mix-and-match different combinations of choices. Precisely, we train the following models (reported in fig. 3, in orange):

- Gemma-2b and Gemma-2b-it fine-tuned on the MCQA datasets (respectively, "MCQA" and "it-MCQA" in the plots)
- Gemma-2b and Gemma-2b-it trained with DPO and fine-tuned on the MCQA datasets ("DPO-MCQA" and "it-DPO-MCQA")
- Gemma-2b-it pre-trained with Orca-Math, with and without DPO, finally tuned for MCQA: "it-ORCA-MCQA" and "it-ORCA-DPO-MCQA"

Table 1 reports the μ-averaged accuracy among test sets. These results clearly show the improvement of fine-tuned models over the baseline, both considering the zero-shot and one-shot case. Moreover, it is noticeable that fine-tuning an instructed model leads to better results. Finally, a negligible improvement is achieved by models that underwent also DPO training. On the other hand, a single

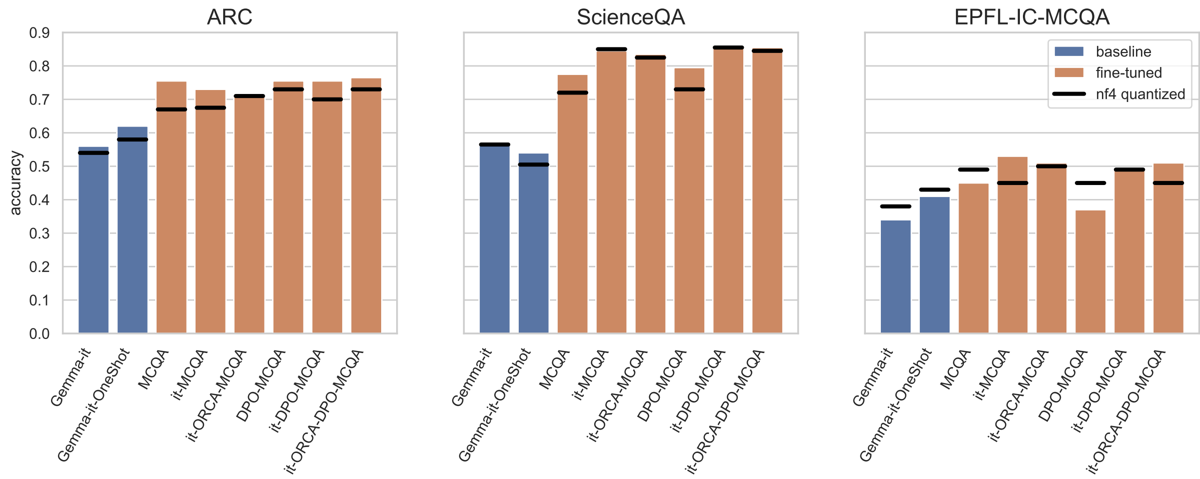

Figure 3: **Accuracy in MCQA task across different test sets**, for different configurations of our training pipeline. The black line for each bar represents the accuracy value obtained after 4-bit quantization. Baseline models are reported in blue, fine-tuned model in orange. Models with "it" in their names were trained starting from Gemma instructed, ORCA refers to further pre-training using Orca dataset and finally MCQA refers to the third step of the trianing pipeline.

DPO training requires almost 2 days of computation, while the MCQA fine-tuning step takes about 30 minutes. So, we conclude that an heavy DPO training does not bring any significant improvement and that it is not worth investing such demanding computational expenses.

Moreover, table 1 also proves that the quantized version of our models generally retain most of the performance on the test sets. From our results, we cannot attribute any role to DPO in retaining performance, as the average performance loss is 0.025 for DPO models and 0.031 for non-DPO ones, when quantized. Notably, the best quantized models are also the best before quantization.

Section 5 reports both a qualitative evaluation and an analysis of the answers generated by our best model, *it-ORCA-DPO-MCQA*, that will constitute our final system STEMERALD .

5 Analysis

In this section we analyze the output of STEMERALD to understand its strengths and weaknesses. We first perform a qualitative evaluation of some sampled answers, focusing on mistakes made by the system and categorizing them using the criteria defined in section 4.2. Also, we present a fine-grained visualization of the model accuracy, splitting the questions by subject (fig. 4). Note that only ScienceQA and EPFL-MCQA dataset provide information about the topic of the question.

5.1 Qualitative Evaluation

We inspected 50 generated answers and reported the relevant ones. Coherently with the accuracy in reported in table 1 roughly 70% of the answer were correct (such as .4), but not always a correct answer means a flawless explanation. The full generated answers are reported in section 7, here we only report some portion of the generations.

First, the model sometimes refers to an option without being clear on which option is the subject of the sentence (as in example .1). We also noticed inconsistency between the explanation and the chosen option .3. Crucially, many generations contain untruthful knowledge, often providing well detailed and precise answers where a single wrong term completely flips the meaning of the answer. For instance, in .6 the model states that the last layers of a neural network contain generic features (while the "first layers" would be correct). It also fails to answer questions related to very simple concepts, such as "by going from underwater toward the surface you get closer to the center of Earth" (see .2).

Curiously enough, sometimes the quantized model performs better than the original one (see .7), but other times it fails (see .5).

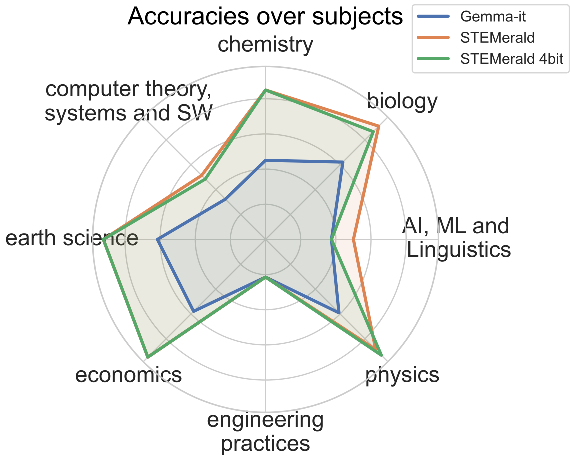

Figure 4: **AI, ML and linguistic is the only field in which the quantized version shows degraded performance** comparable to the non-specialized model (Gemma-it). Note that every tick in the plot corresponds to 20%.

6 Ethical considerations

6.1 Language Adaptation for High-Resource and Low-Resource Languages

Non-English datasets are more rare online, but they could be generated with a similar procedure to the one used to create the EPFL preference data. Specifically, we could request exam question datasets from universities where the primary language is French, German, or Spanish, and utilizing ChatGPT-4 to produce ranked pairs of preferences. In fact, ChatGPT-4 has shown great capabilities for these high-resource languages (Berrezueta-Guzman et al., 2023; Jung et al., 2023), thus it can effectively generate these datasets. The training procedure would then follow the one described in this report.

For low-resource languages, like Urdu and Swahili, the adaptation process presents more challenges due to the scarcity of data. To overcome this issue, different strategies can be employed:

- The development of translation models for low-resource languages (Ranathunga et al., 2023; Gu et al., 2018) would allow translating the dataset used in this project into the low-resource target language. Once translated, the same processes described in the report can be applied to the translated dataset.
- Pre-trained models on high-resource languages with similar linguistic structures can be fine-tuned on low-resource languages. This require a smaller dataset for the final fine-tuning, thus making the procedure feasible even for low-resource languages.

6.2 Interaction with Users in Signed Language

To interact with users in signed language, our model must understand and interpret it. Similarly to what was done by (Lim et al., 2023), we would need to train a vision model (e.g. CNN, Visual Transformer (Dosovitskiy et al., 2020)) to recognize the sign language hands position and facial expression. Several datasets are available for this purpose (Li et al., 2020; Ronchetti et al., 2023). The recognized signs would be translated into natural language and then inputted into our LLM. The rest of the pipeline would remain the same.

6.3 Potential Harms

A well-implemented student assistant system may lead to student over-reliance (Milano et al., 2023). If our teacher assistant works perfectly, students might avoid solving homework and projects independently, resulting in poor learning outcomes. Furthermore, since our model can still make mistakes, students relying on it might learn incorrect information. Therefore, teacher supervision remains crucial for ensuring accurate educational outcomes, and the model itself could be futher tuned to push the students to interact with their teachers.

7 Conclusion

In this project, we demonstrated that STEMERALD , our system trained on top of Gemma-2b, is a powerful tool for STEM education. It achieves notable accuracy in multiple-choice question answering and benefits a massive reduction in memory consumption from quantization, requiring only 2GB of GPU, being suited to operate efficiently on consumer hardware. Despite its specialization to multiple-choice formats, which limits broader application, STEMERALD project sets a promising foundation for future explorations of specializing LLMs for STEM students. Future perspectives include adding mechanisms for retrieval augmented generations or fine-tuning a mixture of many models, specialized on different subjects, still retaining benefits of quantization to keep the system size reasonable, while benefiting from more specialized knowledge.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges.

Jonnathan Berrezueta-Guzman, Laura Malache-Silva, and Stephan Krusche. 2023. Chatgpt-4 as a tool for reviewing academic books in spanish. In *Latin American Conference on Learning Technologies*, pages 384–397. Springer.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Wenhu Chen, Ming Yin, Max W.F. Ku, Yixin Wan, Xueguang Ma, Jianyu Xu, Tony Xia, Xinyi Wang, and Pan Lu. 2023. Theoremqa: A theorem-driven question answering dataset. *ArXiv*, abs/2305.12524.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

David Dewhurst, Hamish Macleod, and Tracey Norris. 2000. Independent student learning aided by computers: An acceptable alternative to lectures? *Computers Education*, 35:223–241.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

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.

Zhanming Jie, Trung Quoc Luong, Xinbo Zhang, Xiaoran Jin, and Hang Li. 2023. Design of chain-of-thought in math problem solving. *ArXiv*, abs/2309.11054.

Leonard B Jung, Jonas A Gudera, Tim LT Wiegand, Simeon Allmendinger, Konstantinos Dimitriadis, and Inga K Koerte. 2023. Chatgpt passes german state examination in medicine with picture questions omitted. *Deutsches Ärzteblatt International*, 120(21-22):373.

Firuz Kamalov, David Santandreu Calonge, and Ikhlaas Gurrib. 2023. New era of artificial intelligence in education: Towards a sustainable multifaceted revolution. *Sustainability*, 15(16):12451.

Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. pages 1–20.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. Huggingface h4 stack exchange preference dataset.

Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469.

Zhenwen Liang, Dian Yu, Xiaoman Pan, Wenlin Yao, Qingkai Zeng, Xiangliang Zhang, and Dong Yu. 2023. Mint: Boosting generalization in mathematical reasoning via multi-view fine-tuning. *ArXiv*, abs/2307.07951.

JongYoon Lim, Inkyu Sa, Bruce MacDonald, and Ho Seok Ahn. 2023. A sign language recognition system with pepper, lightweight-transformer, and llm. *arXiv preprint arXiv:2309.16898*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *ArXiv*, abs/2308.09583.

Kamil Malinka, Martin Peresíni, Anton Firc, Ondrej Hujnák, and Filip Janus. 2023. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? In *Proceedings of the*

2023 Conference on Innovation and Technology in Computer Science Education V. 1, pages 47–53.

Silvia Milano, Joshua A McGrane, and Sabina Leonelli. 2023. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.

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.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):1–37.

Franco Ronchetti, Facundo Manuel Quiroga, César Estrebou, Laura Lanzarini, and Alejandro Rosete. 2023. Lsa64: an argentinian sign language dataset. *arXiv preprint arXiv:2310.17429*.

Teo Susnjak. 2022. Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*.

Kehui Tan, Tianqi Pang, Chenyou Fan, and Song Yu. 2023. Towards applying powerful large ai models in classroom teaching: Opportunities, challenges and prospects. *arXiv preprint arXiv:2305.03433*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Jianxun Wang and Yixiang Chen. 2023. A review on code generation with llms: Application and evaluation. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pages 284–289. IEEE.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

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*.

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.

Zheng Yuan, Hongyi Yuan, Cheng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *ArXiv*, abs/2308.01825.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *ArXiv*, abs/2309.05653.

Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. 2024. Lora land: 310 fine-tuned llms that rival gpt-4, a technical report. *arXiv preprint arXiv:2405.00732*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Appendix 0: AI usage appendix

Apart for the usage of AI-based tools for data collection, labeling and extraction, all discussed in detail in the report, the only tools our group employed is GitHub Copilot. This tool was mainly used to speed up coding, and never relied on for multiple-lines code snippets, so that we constantly supervised it output.

Appendix 1: Model Prompt for Answer Extraction

Few shot prompt used to make GPT-3.5 extract from a given answer to a MCQ the letters corresponding to the chosen answers.

```
You are a helpful assistant for answer extraction. You will receive an open question with multiple choices and a long explanatory answer. Your task is to extract from the long answer the letter (or letters) corresponding to the correct answer (or answers) and output it as a JSON file.

## Example 1
### Input:
### Question: Which of the following scheduler policies are preemptive?
Options:
A. FIFO (First In, First Out)
B. SJF (Shortest Job First)
C. STCF (Shortest Time to Completion First)
D. RR (Round Robin)

### Answer:
SJF (Shortest Job First) and RR (Round Robin) are preemptive scheduler policies.

SJF is preemptive because if a new job arrives that has a shorter processing time than the currently running job, the scheduler will preempt the current job and switch to the new job.

RR (Round Robin) is also preemptive because each process is given a small unit of time to execute before being preempted and moved to the back of the queue. This allows for fair allocation of CPU time among all processes.

### Response:
{
    "correct_choices": ["B", "D"]
}


## Example 2:
### Input:
### Question: Under which condition is an element $x \in \mathbb{Z}_n$ invertible?
Options:
A. $\mathsf{gcd}(x, \varphi (n)) = 1$.
B. $\mathsf{gcd}(x, n-1) = 1$.
C. $\mathsf{gcd}(x, n) = 1$.
D. $\mathsf{gcd}(x, n) \neq 1$.

### Answer:
The correct answer is: '$\mathsf{gcd}(x, n) = 1$.'

Explanation: An element $x \in \mathbb{Z}_n$ is invertible if and only if it is coprime with $n$. In other words, the greatest common divisor (gcd) of $x$ and $n$ must be 1 for $x$ to be invertible in $\mathbb{Z}_n$. This is because if $x$ and $n$ have a common factor greater than 1, then $x$ cannot have a multiplicative inverse in $\mathbb{Z}_n$.
Therefore, the condition for an element $x \in \mathbb{Z}_n$ to be invertible is $\mathsf{gcd}(x, n) = 1$.

### Response:
{
    "correct_choices": ["C"]
}


### Input:
<input-question-here>

### Response:
```

Appendix 2: qualitative Analysis

.1 Unclear selected answer

```
Question: Suppose Beth decides to go on the drop tower. Which result would be a cost?

Options:
A. Beth will save some ride tickets. She needs fewer tickets to go on the drop tower than on the screaming swing.
B. Beth will give up the chance to go on the screaming swing. She would have had more fun on that ride.

Answer: Let's think step by step.

This result is a cost. It involves giving up or spending something that Beth wants or needs:
Beth will give up the chance to go on the screaming swing. She would have had more fun on that ride.
Thus, the correct answer is: B.
```

.2 Gravity goes up

```
You are a helpful assistant for computer science and STEM students. You will receive a multiple-choice question and you should answer correctly.

Question: Complete the statement. Assume that the whale's mass did not change.
The gravitational potential energy stored between the whale and Earth () as the whale swam toward the surface.

Options:
A. increased
B. stayed the same
C. decreased

Answer: Let's think step by step.

True: A Predicted: C
Think about how the distance between the whale and the center of Earth changed.
The bottom of the pond was lower than the edge of the pond. As the whale swam toward the surface, the distance between the whale and the center of Earth decreased. So, the gravitational potential energy stored between the whale and Earth decreased as the whale swam toward the surface.
Thus, the correct answer is: C.
```

.3 Inconsistency between reasoning and answer

```
Question: During digestion, fructose is produced by the breakdown of larger molecules. Which type of molecule could be broken down to increase the body's fructose levels?

Options:
A. proteins
B. vitamins
C. triglycerides
D. carbohydrates

Answer: Let's think step by step.

True: D Predicted: D
During digestion, fructose is produced by the breakdown of larger molecules, such as starches, proteins, and fats. To increase the body's fructose levels, the breakdown of starches, proteins, and fats would be necessary.
Thus, the correct answer is: D. carbohydrates.
```

.4 Good job!

```
Question: Your input is "Distributed Information Systems". Your model tries to predict "Distributed" and "Systems" by leveraging the fact that these words are in the neighborhood of "Information". This model can be:?

Options:
A. Bag of Words
B. Word Embeddings
C. LDA
D. kNN
```

Answer: Let's think step by step.

True: B Predicted: B
The correct answer is:
- Word Embeddings

Explanation:
- Word Embeddings: Word embeddings are numerical representations of words that capture their semantic meaning. By leveraging the fact that the words "Distributed" and "Systems" are in the neighborhood of "Information", the model can predict these words by analyzing the surrounding words in the text.
- Bag of Words: The Bag of Words model is a statistical model that represents text as a document-term matrix. It does not leverage the context of words in the neighborhood of a specific word.
- LDA (Latent Dirichlet Allocation): LDA is a probabilistic topic modeling algorithm that can be used to identify latent topics in a collection of documents. It does not directly leverage the context of words in the neighborhood of a specific word.
- kNN (k-Nearest Neighbors): kNN is a supervised machine learning algorithm that classifies a new data point based on the majority class of its k nearest neighbors. It does not leverage the context of words in the neighborhood of a specific word.

Therefore, the model can be predicted by leveraging the fact that the words "Distributed" and "Systems" are in the neighborhood of "Information".
Thus, the correct answer is: B.

QUANTIZED
True: B Predicted: B
The correct answer is:
- Word Embeddings

Explanation:
- Word Embeddings: Word embeddings are a technique used in natural language processing to represent words as vectors in a high-dimensional space. This technique allows us to capture semantic relationships between words, which can be leveraged to predict the correct word based on the context.
- Bag of Words: The Bag of Words model is a technique used in information retrieval to represent documents as lists of words. It does not leverage the context of words in the neighborhood of "Distributed" and "Systems" to predict the correct word.
- LDA (Latent Dirichlet Allocation): LDA is a probabilistic topic modeling technique used in natural language processing to identify latent topics in a collection of documents. It does not leverage the context of words in the neighborhood of "Distributed" and "Systems" to predict the correct word.
- k-Nearest Neighbors (k-NN): k-Nearest Neighbors is a supervised machine learning algorithm used for classification. It does not leverage the context of words in the neighborhood of "Distributed" and "Systems" to predict the correct word.

Therefore, the correct answer is Word Embeddings.
Thus, the correct answer is: B.

.5 Quantization messed it up

Question: The test loss of the 1-nearest neighbor classifier is always zero.

Options:
A. TRUE
B. FALSE

Answer: Let's think step by step.

True: B Predicted: B
The statement is FALSE. The test loss of the 1-nearest neighbor classifier is not always zero. In fact, the test loss of the 1-nearest neighbor classifier can be non-zero. This is because the 1-nearest neighbor classifier can sometimes misclassify the data point being tested, leading to a non-zero test loss.
Thus, the correct answer is: B.

QUANTIZED
True: B Predicted: A
The statement is true. The test loss of the 1-nearest neighbor classifier is always zero because it

is the simplest classifier that always predicts the class of the training data. The test loss is the loss of the classifier on unseen data, and since the 1-nearest neighbor classifier always predicts the class of the training data, the test loss is always zero.
Thus, the correct answer is: A.

.6 Untruthful

Question: You want to build a convolutional neural network to distinguish between types of cars in images. Your friend Alice, a biologist, has been working on a network to classify wildlife, which she calls WildNet. She spent several weeks training that network, and made it accessible to you. What can you do with it?

Options:
A. Nothing, wildlife is not the same as cars.
B. I can't reuse any of the weights of any layer, but I can take inspiration from the architecture of WildNet.
C. I can freeze the last few layers, and then remove the early layers and replace it with my own re-learned layers. That way, I can make use of the generic features learned by WildNet.
D. I can freeze the early layers, and then remove the last layers and replace it with my own re-learned layers. That way, I can make use of the generic features learned by WildNet.
E. I can use WildNet right away for my task, without modifying anything.

Answer: Let's think step by step.

True: D Predicted: C
The correct answer is: I can freeze the last few layers, and then remove the early layers and replace it with my own re-learned layers. That way, I can make use of the generic features learned by WildNet.

Explanation:
1. Freeze the last few layers of WildNet: This step is crucial as it allows the network to retain the generic features learned by WildNet, which can be reused in your task.
2. Remove the early layers: This step involves removing the layers that were trained on specific types of cars in the WildNet model.
3. Replace the early layers with your own re-learned layers: This step involves retraining the network on your task data to learn new features and make predictions.
4. Use WildNet as a baseline: By freezing the last few layers of WildNet and removing the early layers, you can leverage the generic features learned by WildNet as a starting point for your task.
5. Make use of the generic features learned by WildNet: By replacing the early layers with your own re-learned layers, you can make use of the generic features learned by WildNet to improve the performance of your task.
Thus, the correct answer is: C.

.7 Quantized is better

Question: Categorical Cross-Entropy loss:?

Options:
A. Minimizing the distance between the predicted point and the true point
B. Maximizing the probability of the correct class
C. Minimizing the score of false classes when they are close, or bigger than, the score of the true class
D. Maximizing the accuracy

Answer: Let's think step by step.

True: C Predicted: D
The correct answer is: Minimizing the score of false classes when they are close, or bigger than, the score of the true class

Explanation:
- Categorical Cross-Entropy loss is a loss function used in machine learning to measure the difference between the predicted probabilities of a class and the actual probabilities of a class.
- Minimizing the distance between the predicted point and the true point: This is not the correct answer as the goal of using a loss function is to minimize the distance between the predicted probabilities and the actual probabilities.
- Maximizing the probability of the correct class: This is not the correct answer as the goal of using a loss function is to minimize the distance between the predicted probabilities and the actual probabilities.
- Minimizing the score of false classes when they are close, or bigger than, the score of the true class: This is the correct answer as the goal of using a loss function is to minimize the distance

between the predicted probabilities and the actual probabilities, with the aim of minimizing the score of false classes when they are close to or larger than the score of the true class.
Thus, the correct answer is: D.

QUANTIZED
True: C Predicted: C
The correct answer is: Minimizing the score of false classes when they are close, or bigger than, the score of the true class

Explanation:
- Categorical Cross-Entropy loss is a loss function used in classification tasks where the model predicts the probability of belonging to one class.
- The loss is calculated by comparing the predicted probabilities of the true class and the predicted probabilities of the false classes.
- Minimizing the score of false classes when they are close or bigger than the score of the true class means that the model is more likely to predict the true class correctly.
- This approach helps the model to focus on correctly classifying the true class and minimizes the impact of false predictions.
Thus, the correct answer is: C.