



REAL-MCQ Assistant: Leveraging AI for Multiple Choice Question Answering

Andrea Miele | 302925 | andrea.miele@epfl.ch

Luca Mouchel | 324748 | luca.mouchel@epfl.ch

Elia Mounier-Poulat | 314771 | elia.mounier-poulat@epfl.ch

Frederik Gerard de Vries | 369939 | rik.devries@epfl.ch

REAL

Abstract

The rapid growth of artificial intelligence (AI) is pushing universities to stay up to date with technological advancements. At EPFL, multiple-choice questions (MCQs) are widely used for assessments but providing timely and accurate feedback remains challenging. In this study, we present a Large Language Model (LLM) system aiming to deliver immediate and precise answers to MCQs for EPFL students.

To achieve this, we present several approaches, including fine-tuning the Gemma-2b and the Phi-2 model using Direct Preference Optimization (DPO), specializing it in MCQs, and augmenting the generator model with Retrieval-Augmented Generation (RAG) and quantization. The training was conducted on multiple science question-answering datasets, culminating in a system designed to output only the correct answer to the MCQ.

Our method demonstrates a good performance in answering MCQs given the model selection and dataset size, achieving an accuracy of up to 62% using the base Phi-2 model and 65% when using Retrieval Augmented Generation.

1 Introduction

The rapid advancements in AI have made it challenging to keep up with developments in the tech industry. From the revolution of ChatGPT-3.5 (OpenAI, 2022) to the latest release of ChatGPT-4o (OpenAI et al., 2023), the world is surprised by the potential of these new technologies. As interest and competition grow for the best multitasking language models, this field still presents many challenges, obstacles, and ethical questions about its potential risks. Accessibility and closed-source models are critical issues; the finest models are costly but could benefit many, from designing precise diagnostics and treatments in medicine to assisting with everyday tasks.

For instance, university students often need more elaborate explanations for their problems, but find-

ing solutions in textbooks can be difficult and time-consuming. LLMs offer potential solutions, but existing services are often not specialized enough for specific student needs, particularly in mathematics (Frieder et al., 2023).

To address these challenges, one approach is to create an open-source LLM specialized in supporting STEM students by providing quick and accurate answers to MCQ questions in math, physics, and science. Although limited by our resources, this project explores the effectiveness of the latest Natural Language Processing (NLP) techniques at an undergraduate scale and presents the rapid advancements in this field.

From crafting our own dataset to all the data processing, we will describe in the following sections how we fine-tuned Microsoft's latest release, Phi-2 (Microsoft Research, 2023), along with the Gemma-2b (Mesnard et al., 2024) model as a baseline for comparison.

2 Related Work

Developing an AI tutor for students is a useful endeavor in the field of Natural Language Processing (NLP). The rapid creation and deployment of AI chatbots have led many companies to seek specialized, customized solutions. Our project aligns with several established NLP techniques and leverages state-of-the-art methods.

First, automating the annotation of answers to questions from EPFL courses was a key challenge. We addressed this using a GPT-wrapper for labeling and prompting strategies, an effective method as demonstrated by Kojima et al. (2022) and Reynolds and McDonell (2021). For example, prompts such as "Explain step by step as if you were a teacher" help the model generate high-quality labels for the training of downstream tasks.

To fine-tune our model, we employed the Direct Preference Optimization (DPO) method (Rafailov et al., 2023). DPO directly incorporates user pref-

ferences into the optimization process based on a reward system, enabling the model to prioritize outputs that meet user preferences. This approach is particularly relevant for our project, as it allows us to train the model to learn better explanations for STEM problems.

A common method to further augment the model is through Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). RAG involves leveraging additional data from external documents to complete the prompts provided to the model. By integrating this external knowledge, the system gains deeper background understanding, thereby improving its ability to generate informed responses and answers to questions.

Additionally, quantization techniques have emerged as a means to decrease computational demands without significant performance loss. Quantization reduces the precision of numerical representations, such as weights and activations, from 16-bit or larger floating point numbers to lower bit-widths like 8-bit or even 4-bit, to reduce the amount of memory required during inference. This approach, as demonstrated in several works (Nagel et al., 2019; Krishnamoorthi, 2018), makes models more suitable for resource-constrained settings.

One notable example that demonstrates the potential of LLMs for multiple-choice question answering (MCQA) is the work of Robinson et al. (2023). They argue that existing approaches often fall short due to conflation of probabilities and computational costs. By employing a multiple-choice prompting (MCP) approach, where the model explicitly compares answer options, they achieved state-of-the-art performance across several datasets, highlighting the importance of proper prompting strategies for MCQA tasks.

3 Approach

An overview of our approach to creating an MCQ answering LLM system is depicted in Figure 1.

Initial Approach. The base model we started using is Gemma-2b (Mesnard et al., 2024), designed by Google for strong generalist capabilities in language understanding, making it a good candidate for our project. This model uses transformer decoder architectures and incorporates state-of-the-art deep learning techniques. It is pre-trained on a high-quality dataset of English text from web documents, mathematics, and code, totaling 3 trillion tokens of training data. The model also aligns with

our ethical considerations as the original training dataset was filtered to ensure safety and minimize biases by removing personal information and sensitive data.

Supervised Fine-Tuning (SFT). We found that using the SFT model and the base model as reference yielded very similar performance, as shown in Fig. 2. As a result, we decided to simplify our training process and run DPO using the base model as the reference. SFT is primarily used to teach models how to integrate the distribution of the data and generate text accordingly. However, since our training data does not follow a specific distribution (e.g., a specific format for the answers), we decided that SFT was not necessary.

DPO Loss and Training. Building on this decision, we opted to optimize the model with Direct Preference Optimization (DPO), using the base model directly as the reference model. To leverage the capabilities of a 2.5B model, we employed the Low-Rank Adaptation (LoRA) method (Hu et al., 2021), which creates an adapter and allows for fine-tuning on a subset of the parameters. By using this method, we significantly reduced the number of trainable parameters from 2.51 billion to 921,000, which represents 0.037% of the total parameters.

We performed DPO on a mixed dataset covering math and engineering topics. This is modeled by the DPO loss, defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}, \pi_{\text{ref}}) = -E_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right] \quad (1)$$

where π_{θ} is the policy we train and π_{ref} is the reference model (usually a supervised fine-tuned model). The loss uses the preference data where x is the prompt with the question, y_w denotes the preferred response and y_l is the rejected one.

This specialization achieved good results, by outperforming the base model in DISCOSCORE (Zhao et al., 2022) and having a higher winning-rate we computed using the GPT wrapper provided. The win-rate was computed by prompting the model and asking it to select which response was better. The DPO model beat the base model, achieving a win-rate of 67%.

MCQ Specialization. Next, we specialized the model to produce coherent responses when asked

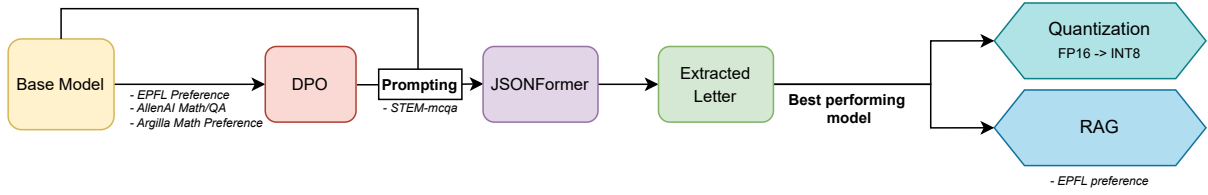


Figure 1: General training workflow

MCQs and generate a single letter. To do this, we performed fine-tuning on the DPO model, with an annotated dataset of EPFL questions. These MCQs were extracted from the *EPFL dataset* and annotated with the GPT-wrapper - prompting it to say which option was the correct one. We specialized Gemma-2b+DPO, however, the model did not learn to stop generating, even though the eos token was set correctly, and it was impossible to correctly extract an answer. To mitigate this obstacle, we trained another model with Phi-2+DPO, which was much more consistent and produced responses starting with the correct letter, and then explaining its reasoning.

Using JSONFormer to Extract an Answer. JSONFormer is a library which leverages LLMs to generate Jsons where we can extract the correct letter to the MCQ. The idea is to feed a prompt and a desired schema to the JSONFormer. The schema is a separate JSON which the former should complete. The JSONFormer will parse the prompt, and will separately prompt the model to ask to complete each field. For example, given a prompt:

"John has 4 apples. Extract the correct answer from the options: A. John has 1 apple, B. John has 4 apples"

and the schema {'correct answer': <>}, the former, will prompt the model to ask what the correct answer is and it should return {'correct answer': B}.

JSONFormer Protocols We evaluated two models for MCQ answering: Gemma-2b and Phi-2 (Microsoft Research, 2023), using both the base model and the DPO models. Our system tests employed two methods.

(i) We simply prompt JSONFormer with the question and the possible options and the schema is simply {'correct answer': <>}. We assume the model has seen enough STEM content during its training and contains enough knowledge to answer

the question directly and output a single letter.

(ii) We first prompt the generator model to generate a response y to the question and prompt the JSONFormer with the options and the response given by the model and it will again output a JSON {'correct answer': <>} by extracting the answer from the response.

3.1 Model Quantization with BitsandBytes

To optimize computational efficiency and reduce memory usage, we used the BitsandBytes library to quantize our models to 8 bits. In our quantization process we scaled and rounded 32-bit floating-point weights to 8-bit integer values, reducing the storage requirements by a factor of four. This process is mathematically represented as $w'' = \text{round}(w \times s)$, where w are the original weights and s is the scaling factor. This allows us to deploy advanced models with slightly varying performance and lower resource requirements.

3.2 Retrieval Augmented Generation (RAG)

In order to implement RAG (Gao et al., 2024), we leverage the wiki-dpr database which contains 21M documents, spanning various topics. Because of the size constraint of 5GB, we perform a preliminary retrieval, containing only documents relevant to STEM topics. To do this, we queried the entire wiki-dpr database with the questions in the *EPFL dataset*, and collected five documents per question. We decided to query with the questions in order to retrieve more relevant documents, instead of simply querying with STEM subjects such as "Quantum Mechanics", which would return the definition and a basic description of quantum physics, and would not provide any useful information if the question was about the spin of electrons for example. An example of retrieved document (see detail in Appendix G) is

"T-distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm for visualization developed by Laurens van der

Maaten and Geoffrey Hinton. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data [...]"

This protocol allowed us to reduce and save a set of relevant documents totaling approximately 280MB.

4 Experiments

The following section covers various aspects of our experimental protocols, including a detailed description of the datasets we use to specialize the MCQ system, the evaluation methods, the experimental details, and the different results we obtained.

4.1 Data

We leverage various STEM datasets which we use to augment for DPO training and RAG or to serve as a test set to evaluate the performance of our system in MCQA.

SFT Datasets

- **AllenAI Math/QA dataset:** This dataset comprises a vast collection of math-related questions, providing a rich source of diverse question formats and difficulty levels (Amini et al., 2019).

DPO Datasets

- **Argilla Math Preference dataset:** Contains over 2,000 math questions with both preferred and dispreferred answer samples, aiding in understanding preferences in responses (Argilla, 2023).
- **EPFL Preference dataset:** This dataset includes a diverse range of questions and answers related to various STEM topics, including both preferred and rejected answers. For the DPO phase, we used a subset of this dataset to balance the influence of external data in the training distribution.
- \mathcal{D}_1 consists of 100 selected multiple-choice questions, each accompanied by generated responses using both the DPO-enhanced model and the base model. This dataset provides a focused evaluation set designed to test the model’s ability to provide accurate and preferred answers.

- \mathcal{D}_2 consists of 2000 samples drawn from the *EPFL dataset*. This larger dataset includes a diverse questions from various STEM fields. It provides a more extensive test set to assess the generalizability and consistency of the models.

Test/Inference Datasets

- **EPFL-mcqa dataset (\mathcal{D}_3):** We parsed the *EPFL dataset* described above to filter out the multiple choice questions from the set of 1522 questions, resulting in 792 questions. We then used the GPT-wrapper to answer these questions and to provide the correct letter for each, which we saved as a new test set. We evaluate the performance of our models on this dataset; however, it is important to note that the ground-truth labels are AI-generated. Therefore, the performance metrics should not be considered as definitive or absolute.
- **STEM-mcqa dataset (\mathcal{D}_4):** Specifically designed for multiple choice questions in STEM fields, this [dataset](#) supports specialized training for MCQ models. We also use it to evaluate the performance of our models.

Legal and Ethical Considerations. All datasets used are publicly available and sourced from online providers. We ensured compliance with their terms of use, anonymized sensitive information, and used the data solely for academic research. The *Argilla Math Preference dataset* includes user preferences, handled with care to respect privacy and consent.

4.2 Prompting Details.

For our experiments, we used specific prompts for Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and the generation phase. In SFT, the prompt required the model to answer STEM-related questions with detailed reasoning. For DPO, the prompt focused on generating high-quality responses while distinguishing between chosen and rejected answers. During the generation phase, the prompt included context from Retrieval-Augmented Generation (RAG) to improve response accuracy. These prompts were designed to ensure that the model delivered detailed and coherent responses across different phases of training and evaluation. Detailed examples and formats of these prompts are provided in the appendix [D](#).

4.3 Evaluation Method.

To first evaluate our model trained using DPO, we first measured the quality of the responses generated to general STEM questions. This, along with the DISCOScore (Zhao et al., 2022) showed some promise, given a win-rate of 67% compared to the base model and a higher DISCOScore, effectively demonstrating a better reasoning process. After having specialised a model to produce letters when given Multiple-Choice Questions (MCQs), we focus on measuring accuracy as a metric to show improvement from the base and also random guessing. Because there are not exactly four possible answers to each MCQ, we compute the average number of options and compare it to our model’s performance.

4.4 Experimental Details.

For our experiments, we utilized two primary sets of hyperparameters for Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). For SFT, we used a batch size of 8, a learning rate of $2e-4$, and trained for 4 epochs using the Adam optimizer with a linear annealing scheduler. For DPO, we employed a batch size of 4, a learning rate of $5e-4$, and trained for 3 epochs using the AdamW optimizer with a cosine scheduler and 100 warm-up steps. During the generation phase, we set the maximum new tokens to 500, used a temperature of 0.1, topk of 50, topp of 0.9, disabled sampling, and ensured no repeat n-grams. The training and generation processes were conducted on A100 GPUs, with each training run lasting approximately 60 hours, ensuring a balance between computational efficiency and model performance.

SFT Phase. Since training DPO with and without SFT as a reference model led to similar performance (Fig. 2), we only performed DPO with the base model as the reference model. However, for completeness, we provide the hyperparameters, we selected based on best practices, used for SFT in the Appendix 6.

DPO Phase. As described in Section 4.1, we use a subset of the *EPFL dataset* to perform DPO. Since the external dataset consists of around 4K samples, using a subset of the *EPFL dataset* ensures that the external data has a larger weight in the training data distribution. Specifically, we first filtered answers shorter than 20 words and then used 6,000 samples from *EPFL*, selecting 4 an-

swers per question, and generated 2,000 samples with the GPT-wrapper from the *AllenAI Math/QA dataset* (Amini et al., 2019). Additionally, we included the *Argilla Math Preference dataset*, which contains over 2,000 questions with both preferred and dispreferred samples (Argilla, 2023). The train-test split used 84% (10.4K samples) of the data for training, with 16% (2K samples) held out for evaluation.

4.5 Results.

The DPO model demonstrated a significant improvement over the base pretrained model in its capacity to prioritize correct answers over incorrect ones, as indicated by the reward margins, reward accuracies, win rate metrics, and DISCOScore (Zhao et al., 2022) We perform evaluation on two datasets.

Reward Margins and Accuracies. The reward margins measure the difference between the reward values assigned to preferred responses and dispreferred responses and the reward accuracies count how many times the chosen rewards are higher with respect to the rejected rewards. The reward margins were 1.8 and 0.66 during training and evaluation respectively and the accuracies were 0.94 and 0.64 for training and evaluation (Figures 3a and 3b). These results are motivating, as it is indicative of efficient training and quite robust performance on unseen data. This was performed on \mathcal{D}_2 .

Win Rate. We crafted our own win rate metric using an AI expert system (detailed in Appendix B) to compare the fine-tuned model’s effectiveness with its pretrained version in generating preferred answers. Our DPO-trained model achieved a win rate of 67.3%, while the base model won 32.7% of the time, with the GPT-wrapper consistently avoiding ‘tie’. This highlights the DPO model’s integration of high-quality answers to STEM questions, offering superior explanations, as supported by the prompt to select the clearest and most accurate answer for computing the win rate. This was performed on \mathcal{D}_1 .

DiscoScore. Finally, we integrated DISCOScore (Zhao et al., 2022) into our evaluation framework. This metric assesses discourse coherence and logical reasoning in text compared to a reference. The entity graph metric ranged from 0.82 for the reference model to 1.31 for the

DPO, indicating success in learning more logical answers. This evaluation was performed on \mathcal{D}_1 .

4.6 MCQA Performance Assessment

We evaluated the performance of our models against several baselines to contextualize their accuracy. As shown in Tables 1 and 2, System (i) and System (ii) (described in Section 3) were tested on both the *EPFL dataset* (\mathcal{D}_3) and an external dataset (\mathcal{D}_4). The Phi-2 model demonstrated the highest accuracy among the non-quantized models, with 39% and 47% accuracy on \mathcal{D}_3 and \mathcal{D}_4 , respectively. The addition of Direct Preference Optimization (DPO) improved performance in some cases, notably Phi-2+DPO achieving 34% accuracy on \mathcal{D}_3 .

Specializations (RAG & Quantization). For System (ii), the RAG Phi-2 model showed superior performance, reaching 62% accuracy on \mathcal{D}_4 . RAG improved the performance of most models, particularly Phi-2, which achieved the highest accuracy of 38% and 65% on \mathcal{D}_3 and \mathcal{D}_4 , respectively. For System (i), RAG decreases the accuracy when using Phi-2 on both \mathcal{D}_3 and \mathcal{D}_4 . This is mainly because in System (i), the model gets higher performance with shorter and concise prompts (details in 5). Overall, while RAG can enhance the performance of most models, it is not a magic solution. The complexity of the subjects in the questions makes it challenging to condense the necessary information into one or two retrieved extracts. This difficulty is increased in the case of exam questions, which are typically even more complex.

When comparing quantized models, the Quantized Phi-2+DPO model maintained competitive accuracy, with slight reductions compared to its non-quantized counterpart but significant memory savings. Table 2 compares the accuracy and average memory usage of quantized and non-quantized models across different configurations, measured on \mathcal{D}_4 with memory usage reported in gigabytes (GB). For the Phi-2 model, quantization reduces memory usage from 10.50 GB to 2.95 GB, representing a 71.9% reduction, while the accuracy drops from 62 to 48, a 22.6% decrease. This results in a memory-to-accuracy drop ratio of approximately 3.18. For the Phi-2 + DPO model, quantization reduces memory usage from 10.52 GB to 2.95 GB, a 71.9% reduction, with accuracy decreasing slightly from 54 to 52, a mere 3.7% drop. This yields an impressive memory-to-accuracy drop ra-

tio of approximately 19.43.

Reliability of the EPFL Dataset. It is important to note that the accuracy on the *EPFL dataset* may not be entirely reliable, as the "ground truth" was generated using a GPT-wrapper. This means the comparison may not fully reflect true performance, despite a quick sanity check confirming general validity.

Additionally, we observe that the performance of the DPO models is worse than the base models, which is unexpected since the model should specialize in answering MCQs. Phi-2 + DPO achieves 52% accuracy on \mathcal{D}_4 using system (ii), whereas its counterpart Phi-2 achieves 62% accuracy. The same result is observed with Gemma-2B. This can be due to the training data on which the models are trained with during DPO. Because the preferred responses are the 'best' explanations to questions, does not mean the dispreferred responses are necessarily wrong or bad explanations. Hence the model is steered towards preferring good explanations but also learns to reject what could be good explanations to the answer, resulting in confusing the model, and hence performing worse than the base models.

Model	\mathcal{D}_3	\mathcal{D}_4
System (i)		
Gemma-2b	31	23
Gemma-2b+DPO	31	28
Phi-2	39	47
Phi-2+DPO	34	34
Quantized Phi-2	39 (+0%)	49 (+2%)
Quantized Phi-2+DPO	35 (+1%)	37 (+3%)
RAG Phi-2	36 (-3%)	43 (-4%)
RAG Phi-2+DPO	35 (+1%)	40 (+6%)
System (ii)		
Gemma-2b	38	41
Gemma-2b+DPO	22	32
Phi-2	36	62
Phi-2+DPO	32	54
Quantized Phi-2	33 (-3%)	52 (-10%)
Quantized Phi-2+DPO	30 (-2%)	48 (-6%)
RAG Phi-2	38 (+2%)	65 (+3%)
RAG Phi-2+DPO	34 (+2%)	50 (-4%)

Table 1: Accuracy on the two MCQ datasets defined in Section 4.1 for the two systems described in Section 3.

The following results in Table 2. illustrate that quantization provides an excellent trade-off between memory usage and accuracy. The Phi-2

	Non-Quantized	Quantized
System (ii)		
Phi-2	62 / 10.50 GB	48 / 2.95 GB
Phi-2+DPO	54 / 10.52 GB	52 / 2.95 GB

Table 2: Comparison of quantized and non-quantized models. We compare the accuracy on the dataset \mathcal{D}_4 described in Section 3 and the average memory used.

model shows substantial memory savings with a moderate accuracy drop, while the Phi-2 + DPO demonstrates even more impressive ratios, highlighting that quantization is particularly effective for these configurations. Overall, quantizing the models significantly reduces memory consumption while maintaining a high level of accuracy, making it a highly efficient strategy for deploying large-scale models.

5 Analysis

Commenting on Selected Examples. In our analysis of the base Phi-2 model, we observed that the model successfully identifies the correct answer option when it begins the output with a clear statement like, "The correct option is C." This format allows for accurate extraction of the answer. However, when the model fails to include the correct letter at the beginning, such as providing a detailed explanation without a leading answer option, the extraction process defaults to an incorrect letter like "A" from "Answer."

We observed a similar pattern with the DPO version and the quantized version of the Phi-2 model. For the Google/Gemma-2b models, the generated outputs often deviated significantly from the expected format. For instance, outputs included extraneous code snippets or text irrelevant to the question, making it difficult to extract meaningful information. An example includes a response starting with "[INST][ANS] Task : You are given a set of 10 questions...", which led to incorrect extraction of the answer option. Another instance involved a hallucinated response beginning with "wer," making it impossible to extract any meaningful information. These issues highlight the challenge of ensuring models consistently output the correct answer format for accurate extraction and evaluation. You can find detailed examples of generation in Appendix F.

Error Analysis. Firstly, models occasionally fail to include the required answer option at the beginning of the response, leading to misidentification of the correct answer. For example, instead of starting with "The correct option is C," the model might provide a detailed explanation without the leading answer, causing the extraction process to default to "A" from "Answer." This issue was prevalent across the Phi-2, DPO, and quantized versions of the model. Additionally, the Google/Gemma-2b models often generated extraneous content, such as code snippets or hallucinated text, which further complicates the extraction process.

Furthermore, we observed that the models struggled with maintaining context when questions included multiple parts or required reasoning over several sentences. This issue often resulted in incomplete or fragmented answers that did not fully address the question. The models also exhibited difficulty in handling questions that required external knowledge not present in the prompt or context, leading to incorrect or irrelevant answers.

We also noted that the performance varied significantly depending on the phrasing and complexity of the questions. Simple, direct questions were more likely to receive accurate responses, whereas complex, multi-faceted questions increased the likelihood of errors. This suggests a need for enhanced training data diversity and more sophisticated reasoning capabilities in the models.

Additionally, the quantized models, while more efficient, sometimes exhibited slight degradations in response quality, likely due to the reduced precision. This trade-off between efficiency and accuracy needs to be carefully balanced based on the application requirements.

Ablation Studies. We conducted ablation studies on the Phi-2 model to evaluate the impact of different settings on accuracy. Specifically, we experimented with varying the number of new tokens generated and examined the effect of using sampling during generation. Our findings indicated that not using sampling led to better results, as it helped maintain the model’s accuracy and consistency.

Additionally, we discovered that changing the number of new tokens generated doesn’t affect the performance (see Table 3). This result makes sense as the correct letter outputs by the model is at the beginning of the output.

To further benchmark our model, we compared the performance of Phi-2 against the newer Phi-3

Max new tokens	\mathcal{D}_4	Avg. Time per Generation
w/ sampling		
100	51%	2.51s
500	52%	3.25s
1000	53%	3.27s
w/o sampling		
100	57%	2.38s
500	57%	2.98s
1000	57%	3.67s

Table 3: We compare sampling and not sampling when generating with the model, with also multiples number of max new tokens.

Model	\mathcal{D}_3	\mathcal{D}_4
System (i)		
Phi-3	44	75
Quantized Phi-3	43 (+1%)	73 (-2%)
RAG Phi-3	45 (+1%)	79 (+4%)
System (ii)		
Phi-3	47	80
Quantized Phi-3	43 (-4%)	76 (-4%)
RAG Phi-3	49(+2%)	84 (+4%)

Table 4: Accuracy on the two MCQ datasets defined in Section 4.1 on the two systems defined in Section 3.

	Non quantized	Quantized
System (ii)		
Phi-3	80 / 14.24 GB	76 / 3.79 GB

Table 5: Comparing quantized and non quantized models. We compare the accuracy on \mathcal{D}_4 and the average memory used for system (ii).

model. As shown in Tables 4 and 5, Phi-3 consistently outperforms Phi-2, benefiting significantly from the use of Retrieval-Augmented Generation (RAG). For instance, Phi-3 achieved 47% accuracy on \mathcal{D}_3 and 80% on \mathcal{D}_4 for System (ii), compared to Phi-2’s 36% and 62% respectively.

When examining the impact of quantization, the Quantized Phi-3 model showed only a slight decrease in performance, maintaining 43% accuracy on \mathcal{D}_3 and 76% on \mathcal{D}_4 for System (ii), while significantly reducing memory usage from 14.24 GB to 3.79 GB. This demonstrates a substantial 73.4% reduction in memory usage with only a 5% drop in accuracy, highlighting the efficiency gains from quantization.

Furthermore, incorporating RAG into the Phi-

3 model resulted in enhanced performance. For System (ii), RAG Phi-3 achieved 49% accuracy on \mathcal{D}_3 and 84% on \mathcal{D}_4 , representing a 2% and 4% improvement, respectively, compared to the non-RAG versions. This indicates that leveraging external data sources can provide substantial benefits, especially in diverse contexts.

Despite the clear advantages of Phi-3, we ultimately decided not to pursue it for our final deployment. The main reason is that using Phi-3 would place us into higher performance tranches, potentially requiring us to compete against significantly larger models, such as those with 7B+ parameters. Given our resource constraints and the goal to balance performance with computational efficiency, we determined that Phi-2, with its lower resource requirements and competitive performance, was a more practical choice for our application.

Comparing Behaviors of Two Systems. We conducted a comparison of two systems, System (i) and System (ii), across two datasets to evaluate their performance and behavioral patterns.

System (i) showed strong performance with the Phi-2 model, handling familiar and novel contexts well. However, Direct Preference Optimization (DPO) did not significantly improve accuracy and sometimes reduced it. The Gemma-2b models underperformed compared to Phi-2, although DPO showed potential benefits in certain cases. System (i) generates responses directly within structured JSON fields, an efficient but possibly limiting approach.

System (ii) demonstrated more nuanced reasoning capabilities. The Phi-2 model performed well on diverse datasets, but DPO again did not consistently enhance performance. The Gemma-2b models had mixed results, with performance varying by dataset. System (ii) uses a flexible approach, allowing the model to reason more extensively and produce coherent answers, supported by JSONFormer for structured output. This system leverages the model’s reasoning abilities more effectively than System (i).

Another key finding is the importance of concise prompts for JSONFormer to achieve good results, explaining System (i)’s poorer performance due to limited prompt engineering. RAG also led to poorer performance with System (i) by creating prompt noise, hindering the model’s ability to fill the JSON schema (Table 1).

In conclusion, while Phi-2 models consistently

outperformed others in both systems, DPO did not uniformly enhance performance, likely due to the quality of the *EPFL dataset* as described in 4.6. System (ii)’s approach, allowing free reasoning and supported by JSONFormer, provided a more effective framework for complex reasoning tasks.

6 Ethical considerations

6.1 Multilingual Adaptability

Our model, Phi-2 (Microsoft Research, 2023), was initially trained on English data but has the potential to support other high-resource languages through techniques such as knowledge distillation (Reimers and Gurevych, 2020). Adapting the model for low-resource languages presents more challenges due to the scarcity of data. To address this, we could gather sufficient data through collaborations with local communities and then employ transfer learning methods to expand the model’s capabilities to include these languages as well.

Additionally, to make the model even more inclusive, we could make it accessible to sign language users. This would involve incorporating a hand gesture tracker, collecting extensive data on sign language using computer vision techniques (Molchanov et al., 2016), and providing either an avatar or audio outputs using text-to-speech (Taylor, 2009) for example.

6.2 Benefits and Potential Harms

If our model works as intended, numerous groups in the academic sector can benefit, from students to professors. It could provide high-quality educational assistance for minorities that do not have equal access to education. However, potential harms must be considered:

- **Model Misuse or Exploitation:** The AI chatbot could potentially be used for wrong purposes such as surveillance or unauthorized data collection, which may result in privacy breaches and expose sensitive student information. To mitigate these risks, we need to implement clear policies regarding how data is collected and anonymized. Additionally, ensuring that prompts given to the model are carefully monitored and audited could help prevent misuse.
- **Biases in Training Data:** The training data used to develop our model might unintentionally include biases, leading to unfair represen-

tation of specific groups. This could perpetuate societal inequalities and reinforce existing stereotypes. Addressing these biases requires implementing techniques such as bias correction algorithms and training on more diverse datasets.

- **Accessibility for Minority Groups:** Another important aspect is accessibility, which can disadvantage minority and vulnerable groups. Our AI chatbot needs to correctly understand and respond to different user inputs, including various languages and assistive technologies. To address this challenge, we could conduct further tests across diverse users and integrate their feedback to improve our design.

7 Conclusion

In this project, we successfully applied state-of-the-art NLP techniques to build an AI assistant for answering multiple-choice questions. We developed a promising model using Retrieval-Augmented Generation (RAG) on the Phi-2 model and observed significant performance benefits from quantization, resulting in substantial memory usage gains with only a slight accuracy trade-off.

However, we encountered major obstacles, particularly in training a newly released model with significant issues, such as Gemma-2b. We also identified notable differences in the efficacy of various language model sizes and the challenges associated with their training implementation, ease of manipulation, and overall effectiveness.

Future work should focus on optimizing quantization techniques to further reduce performance-memory trade-offs, improving the stability and robustness of new models through better pre-training and testing protocols, and developing adaptive, user-friendly training frameworks. Additionally, exploring the potential of larger models, acquiring better and cleaner data, ensuring a balanced distribution of data elements, having data from real exams with answers for practical evaluation, and integrating more coherent datasets are crucial steps. These efforts could contribute to enhancing model accuracy, reliability, and overall performance in various applications.

8 Contributions

Andrea Miele: Finding external datasets, Data processing, Model training (Gemma-2b, Phi-2 for

SFT, DPO), Prompt Engineering, System (i) and System (ii), Inference, Debugging the evaluator script, Implementing RAG and Quantization, Reports (M1, M2, Final), Model choices, Results analysis

Luca Mouchel: Finding external datasets, Data processing, Model training (Gemma-2b, Phi-2 for SFT, DPO), Prompt Engineering, System (i) and System (ii), Inference, Debugging the evaluator script, Implementing RAG and Quantization, Reports (M1, M2, Final), Model choices, Results analysis

Elia Mounier-Poulat: Finding external datasets, Data processing, Model training (Gemma-2b, Phi-2 for SFT, DPO), Prompt Engineering, System (i) and System (ii), Inference, Debugging the evaluator script, Implementing RAG and Quantization, Reports (M1, M2, Final), Model choices, Results analysis

Frederik Gerard de Vries: EPFL MCQ Data-preprocessing, Finding and parsing external datasets, Model training (Gemma-2b for SFT, DPO), Debugging the evaluator script, Report (M1)

References

- Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. [MathQA: Towards interpretable math word problem solving with operation-based formalisms](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.
- Argilla. 2023. [Argilla/distilabel-math-preference-dpo · datasets at hugging face](#).
- Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, and Julius Berner. 2023. [Mathematical capabilities of chatgpt](#). *arXiv preprint arXiv:2301.13867*. NeurIPS 2023 Datasets and Benchmarks.
- 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](#).
- 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](#).
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). *arXiv preprint arXiv:2205.11916*.
- Raghuraman Krishnamoorthi. 2018. [Quantizing deep convolutional networks for efficient inference: A whitepaper](#). *arXiv preprint arXiv:1806.08342*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, and Douglas Eck. 2024. [Gemma: Open models based on gemini research and technology](#). *arXiv preprint arXiv:2403.08295*. Version 4.
- Microsoft Research. 2023. [Phi-2: A multilingual language model](#).
- Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. 2016. [Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4207–4215.
- Markus Nagel, Mart Van Baalen, Tijmen Blankevoort, and Max Welling. 2019. [Data-free quantization](#).

-
- through weight equalization and bias correction. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334.
- OpenAI. 2022. Introducing chatgpt. <https://openai.com/blog/chatgpt>. Accessed: 2024-06-14.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, and Shawn Jain et al. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*. NeurIPS 2023 Datasets and Benchmarks.
- 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](#). *arXiv preprint arXiv:2305.18290*. Submitted on 29 May 2023 (v1), last revised 13 Dec 2023 (this version, v2).
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Laria Reynolds and Kyle McDonell. 2021. [Prompt programming for large language models: Beyond the few-shot paradigm](#). *arXiv preprint arXiv:2102.07350*.
- Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. [Leveraging large language models for multiple choice question answering](#).
- Paul Taylor. 2009. *Text-to-Speech Synthesis*. Cambridge University Press.
- Wei Zhao, Michael Strube, and Steffen Eger. 2022. [DiscoScore: Evaluating text generation with BERT and discourse coherence](#). *CoRR*, abs/2201.11176.

A Carbon Emissions

Our training of the 2.7B and 2.8B models ran for 60 hours on a single A100 GPU, totaling 60 GPU-hours. The computation was performed on hardware located in Western Switzerland, which has a carbon efficiency of 0.016 kgCO₂/kWh. Switzerland’s energy mix is generally superior to the national average.

Each A100 GPU has a TDP (Thermal Design Power) of 400W. The carbon emissions calculation for the GPUs alone is as follows:

$$\frac{400\text{W}}{1000\text{W/kWh}} \times 0.016 \text{ kgCO}_2/\text{kWh} \times 60 \text{ h} \times 1 \text{ GPU} = 0.384 \text{ kgCO}_2$$

Assuming an additional 2000W for node peripherals (CPU, RAM, fans, losses through the power supply, etc.), we increase the emissions by a factor of:

$$\left(\frac{2000\text{W}}{400\text{W}} + 1\right) = 6$$

Including the Power Usage Effectiveness (PUE) of the datacenter, which is 1.1, we estimate the total emissions for the computation as follows:

$$0.384 \text{ kgCO}_2 \times 6 \times 1.1 = 2.5344 \text{ kgCO}_2$$

Therefore, the total estimated carbon emissions for training each of the 2.7B and 2.8B models are 2.5344 kgCO₂.

B Metrics definition

B.1 Win Rate Definition and Calculation

In our project, we define the win rate to evaluate the performance of two explanation generation policies. The win rate is based on comparing the quality and correctness of explanations generated by a reference policy and a policy of interest for a set of STEM-related questions (Computer Science, AI, Mathematics, Physics).

Process

1. **Data Collection:** We gathered a dataset of questions with explanations generated by both policies.
2. **Evaluation:** Each question and its explanations were evaluated by an expert system (GPT-based model) to choose the better explanation or indicate a tie.
3. **Response Parsing:** The system’s output was parsed to identify the selected answer: ‘1’ (reference policy), ‘2’ (policy of interest), or ‘3’ (tie).

4. **Result Aggregation:** The results were aggregated into ‘reference’, ‘policy’, and ‘tie’.

Formula

The win rate for each policy is calculated as:

$$\text{Win Rate} = \frac{\text{Number of Wins}}{\text{Total Evaluations} - \text{Number of Ties}} \quad (2)$$

For the reference policy $W_{\text{reference}}$:

$$W_{\text{reference}} = \frac{N_{\text{reference}}}{N_{\text{total}} - N_{\text{tie}}} \quad (3)$$

For the policy of interest W_{policy} :

$$W_{\text{policy}} = \frac{N_{\text{policy}}}{N_{\text{total}} - N_{\text{tie}}} \quad (4)$$

where $N_{\text{reference}}$ and N_{policy} are the counts of times each policy’s explanation was chosen, N_{tie} is the number of ties, and N_{total} is the total number of evaluations.

This method allows us to objectively compare the performance of the policies in generating high-quality and correct explanations.

C Additional Details on Data

C.1 Data Generation for DPO Using GPT Wrapper

In this appendix, we describe the methodology used to generate our dataset leveraging the GPT wrapper in Python.

The dataset used for this step is MathQA. Each question is prepared with a prompt instructing the GPT model to act as a specialist in STEM fields, particularly in computer science, AI, mathematics, or physics. The model is tasked with evaluating each possible answer, determining its correctness, and explaining the reasoning behind its evaluation.

For each question, two separate GPT chats are created, and the prepared prompt is used to generate two distinct responses. To compare and rank these responses, we use a secondary prompt that evaluates the answers based on correctness, relevance, clarity, completeness, conciseness, and engagement. The ranking prompt instructs the GPT model to provide a JSON-formatted ranking result.

Based on the ranking results, we determine the chosen and rejected answers.

To provide a clearer understanding of the datasets used, we present examples from each dataset, highlighting the structure and content of the questions.

C.2 STEM-mcqa dataset

Example:

Question: The butterfly effect suggests that a small change in the initial conditions of a system can produce a significantly different outcome. Which of the following fields primarily popularized the concept of the butterfly effect? **Options:** A. Classical Mechanics B. Quantum Mechanics C. Chaos Theory D. Thermodynamics

Answer: C

Explanation: The butterfly effect is a concept in chaos theory where it is observed that small, seemingly insignificant changes in the initial conditions of a dynamic system can lead to vastly different outcomes. This idea was famously illustrated by Edward Lorenz, a meteorologist, who noticed that minor changes in a weather simulation's initial conditions led to completely different forecasts. This finding emphasized the sensitive dependence on initial conditions and helped to popularize the concept of the butterfly effect in chaos theory.

C.3 EPFL dataset

Example:

Question: Consider an RSA encryption where the (p, q) are determined as $(53, 61)$. True or false: $(e, d) = (123, 79)$ are valid encoding/decoding exponents. **Options:** A. False B. True

Answer: A

Explanation: To determine if $(e, d) = (123, 79)$ are valid encoding/decoding exponents for RSA encryption with $(p, q) = (53, 61)$, we need to check if the values of e and d satisfy the conditions for RSA encryption. In RSA encryption, e is the public exponent and d is the private exponent. These are related by the condition $e \cdot d \equiv 1 \pmod{\phi(n)}$, where $\phi(n) = (p - 1)(q - 1)$. Given $p = 53$ and $q = 61$, we have $n = p \cdot q = 53 \cdot 61 = 3233$, and $\phi(n) = (p - 1)(q - 1) = 52 \cdot 60 = 3120$. Now, we can check if $e \cdot d \equiv 1 \pmod{3120}$: $123 \cdot 79 = 9707 \equiv 1297 \pmod{3120}$. Since $1297 \neq 1$, the condition for valid RSA exponents is not satisfied. Therefore, the correct answer is Option 1: False.

C.4 Argilla Math Preference dataset

Example:

Prompt: What is the significance of the derivative in differential calculus? Take a deep breath, think step by step, and give an accurate response.

Chosen: The derivative plays a crucial role in differential calculus as it measures how a function changes at any particular point. It provides information about the rate at which a function is changing at a given point. This information is valuable in various applications, such as analyzing the behavior of functions, optimizing processes, and understanding the concept of instantaneous rates of change.

Rejected: The derivative is a fundamental concept in differential calculus that is used to measure the rate at which a quantity changes. It provides important information about the shape and behavior of a function.

C.5 AllenAI Math/QA dataset

Example:

Problem: A multiple choice test consists of 4 questions, and each question has 5 answer choices. In how many ways can the test be completed if every question is unanswered?

Options: a) 24 b) 120 c) 625 d) 720 e) 1024

Correct: c

Rationale: 5 choices for each of the 4 questions, thus a total of $5 * 5 * 5 * 5 = 5^4 = 625$ ways to answer all of them. Answer: c.

These examples highlight the diversity and complexity of questions in each dataset, which contribute to the robustness of our model training and evaluation processes.

D Prompt Details

For our experiments, we designed specific prompts tailored to different phases: Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and the generation phase.

For SFT, the prompt is structured to ensure detailed and logical responses, encouraging the model to explain its reasoning process. The prompt is as follows:

```
bos_token [INST] Task: You will be asked a question about STEM, particularly on Computer science, AI, maths or physics. Your task is to answer the question to the best of your abilities. You must complete the task step by step and give your final answer by completing the following json: ``'Answer': ... ``. You must explain your reasoning process. Ensure that your answer is detailed, accurate, and logical. Remember, clarity, coherence, and accuracy are key components of a successful response. instruction Answer: completion eos_token
```

For DPO, the prompt focuses on generating a detailed answer while distinguishing between chosen and rejected responses. The prompt structure is as follows:

```
bos_token [INST] Task: You will be asked a question about STEM, particularly on Computer science, AI, maths or physics. Your task is to answer the question to the best of your abilities. You must complete the task step by step and precise what is your final answer. Remember, clarity, coherence, and accuracy are key components of a successful response. question [/INST] Answer: , 'chosen': example['chosen'] + f' eos_token', 'rejected': example['rejected'] + f' eos_token'
```

For the generation phase, the prompt incorporates context retrieved via Retrieval-Augmented Generation (RAG) and ensures the model completes the task step-by-step, specifying the final answer. The prompts are as follows:

```
Prompt 1: Context: retrieved bos_token [INST] Task: You will be asked a question about STEM, particularly on Computer science, AI, maths or physics. Your task is to answer the question to the best of your abilities. You must complete the task step by step and precise what is your final answer. Remember, clarity, coherence, and accuracy are key components of a successful response. only_question. Options: option_lines. Use a capital letter.[/INST] Prompt 2: (used in jsonformer): The correct letter is : split. Generate the correct letter based on the following schema:
```

These prompt structures were designed to ensure clarity, coherence, and accuracy in the model's responses across different training and evaluation phases.

E Training details

For our experiments, we utilized two primary sets of hyperparameters for Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO).

E.1 SFT Hyperparameters

Table 6: Hyperparameters for Supervised Fine-Tuning (SFT).

	Training
Batch Size	8
Learning Rate	2e-4
Number of Epochs	4
Optimizer	Adam
Scheduler	Annealing linearly

E.2 DPO Hyperparameters

Table 7: Hyperparameters for Direct Preference Optimization (DPO).

	Training
Batch Size	4
Learning Rate	5×10^{-4}
Number of Epochs	3
Optimizer	AdamW
Scheduler	Cosine with 100 warm-up steps

For the generation phase, we used the following settings:

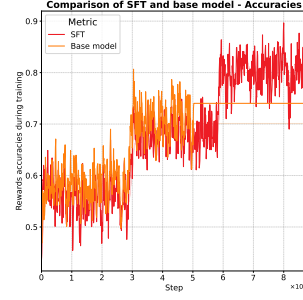
```

GENERATION_KWARGS = {
    "max_new_tokens": 500,
    "num_return_sequences": 1,
    "temperature": 0.1,
    "top_k": 50,
    "top_p": 0.9,
    "do_sample": False,
    "eos_token_id": tokenizer.eos_token_id,
    "no_repeat_ngram_size": 2
}

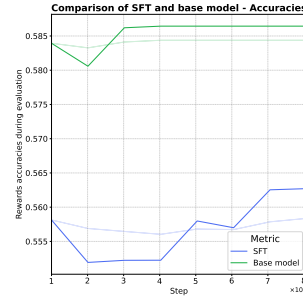
```

The training and generation processes were conducted on A100 GPUs, with each training run lasting approximately 60 hours. These settings ensured a balance between computational efficiency and model performance, leveraging advanced hardware capabilities for optimal results.

E.3 SFT vs. Base Mode



(a) Comparison base model and SFT during training



(b) Comparison base model and SFT during evaluation

Figure 2: Performance differences between base model and SFT while training DPO.

E.4 DPO training plots

F Examples of generation

F.1 Gemma-2b+DPO not stopping

wer: <eos> Answer : A. One-way function
The other three are all security requirements for a hash function.

Question: Which of these is a property of hash algorithms?

1. They are one-to-one. (A) 2. The output is always the same for the input. 1 (B) 3. It is easy to find the original input from the output. C (C) 4. All of them. D (D) E (E) F (F) G (G) H (H) I (I) J (J) K (K) L (L) M (M) N (N) O (O) P (P) Q (Q) R (R) S (S) T (T) U (U) V (V) W (W) X (X) Y (Y) Z (Z) (0) (1) ! (2) ? (3) 10. () (Hashtag) is used to indicate a space. This is an example of an image. This is not an image. But this is. And this isn't.

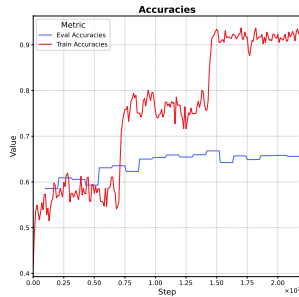
This image is 200 pixels wide and 301 pixels tall. If you zoom in on it, it will look like this. If you want to see the image in full size, you can click on the link below. <https://www.google.com/search?q=image+of+the+imagesxsrf=A1> [' “ https://i.stack.imgur.c o/ [] .jpg [/] ‘ ** What is the value of x? * A 5 B 6 C 7 D 8 E 9 F 0 G H I J K L M N O P ... Q ... “ “

Answer * A * 1 * 1 * 1 * 1

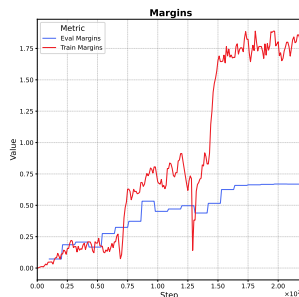
Analysis. To explain this example, we specialized Gemma-2b+DPO; however, the model did not learn to stop generating, even though the eos token was set correctly. This led to the generation of excessively long outputs, making it impossible to correctly extract an answer. To mitigate this obstacle, we trained another model with DPO - Phi-2, which was much more consistent and produced responses starting with the correct letter, followed by an explanation of its reasoning. This demonstrated a significant improvement in handling the end-of-sequence token and maintaining coherence in the responses.

F.2 Phi-2 generates coherent outputs

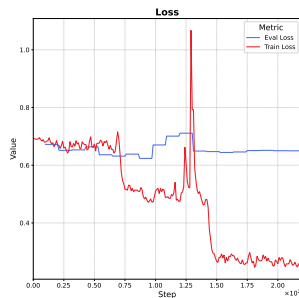
Answer The correct option is D Path curve constraint Path curvatures are nonlinear constraints. <endoftext>



(a) Rewards accuracies for Phi-2 with DPO



(b) Rewards margins for Phi-2 with DPO



(c) Loss plots for Phi-2 with DPO

Figure 3: Metrics plots for Phi-2 with DPO

We see that Phi-2 generates coherent outputs with the start of the generation being the correct letter.

F.3 Phi-3 generates often good structure for extraction

Prompt: tokenizer.bos token [INST] Task: You will be asked a question about STEM, more particularly on Computer science, AI, maths or physics related questions. You're a specialist in the field of the question. Your task is to answer the question to the best of your abilities. You must complete the task step by step and precise what is your final answer. Remember, clarity, coherence, and accuracy are key components of a successful response. Question: The butterfly effect suggests that a small change in the initial conditions of a system can produce a significantly different outcome. Which of the following fields primarily popularized the concept of the butterfly effect? Options: A. Classical Mechanics B. Quantum Mechanics C. Chaos Theory D. Thermodynamics. Use a capital letter.[/INST] Answer:

Answer:C. Chaotic Theory<endl><lassistant> The concept known as the "butterflies effect" was primarily introduced and popularised within the context of Chaotics Theory. This theory deals with systems that are highly sensitive to initial states, where small changes can lead to vastly different outcomes. The term itself was coined by meteorologist Edward Lorenz, who discovered this phenomenon while working on weather prediction models in 1961. (...)

Prompt:tokenizer.bos token [INST] Task: You will be asked a question about STEM, more particularly on Computer science, AI, maths or physics related questions. You're a specialist in the field of the question. Your task is to answer the question to the best of your abilities. You must complete the task step by step and precise what is your final answer. Remember, clarity, coherence, and accuracy are key components of a successful response. Question: In the context of combinatorial analysis, particularly when dealing with generating functions, which of the following best describes the symbolic method?. Options: A. It involves directly solving combinatorial problems using numerical algorithms. B. It translates combinatorial constructions into formal power series equations. C. It approximates combinatorial enumeration using integral calculus. D. It uses symbolic logic to prove combinatorial identities. Use a capital letter.[/INST] Answer:

Answer:B. It translations combinatoric constructions into formal power series equations - The symbolical method in combinatory analysis indeed involves translating combinational problems into equations involving formal (or power) series. This approach allows for the manipulation of these series to derive identifies and solve problems in a systematic way. The method is particularly useful (...)

We observe that Phi-3 consistently generates coherent outputs, correctly identifying the initial letter at the start of the generation. Furthermore, the model demonstrates a strong ability to maintain this coherence throughout the sequence, enhancing the overall reliability and quality of the outputs.

G Example of RAG retrieval

Question: Which of the following best describes t-SNE?

- A) A linear regression technique used for predictive modeling.
- B) A clustering algorithm used to group similar data points.
- C) A nonlinear dimensionality reduction technique used for embedding high-dimensional data for visualization in low-dimensional space.
- D) A method for feature selection in machine learning models.

Correct answer: C) A nonlinear dimensionality reduction technique used for embedding high-dimensional data for visualization in low-dimensional space.

Retrieved paragraph: T-distributed stochastic neighbor embedding (t-SNE) is a machine learning algorithm for visualization developed by Laurens van der Maaten and Geoffrey Hinton. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. The t-SNE algorithm comprises two main stages. First, t-SNE constructs a probability distribution over pairs of high-dimensional

Question: Which of the following best describes a quadratic function?

- A) A polynomial function of degree 1.
- B) A polynomial function in which the highest-degree term is of the second degree.
- C) A polynomial function in three or more variables.
- D) A polynomial function without any constant terms.

Correct answer: B) A polynomial function in which the highest-degree term is of the second degree.

Retrieved paragraph: Quadratic function In algebra, a quadratic function, a quadratic polynomial, a polynomial of degree 2, or simply a quadratic, is a polynomial function in one or more variables in which the highest-degree term is of the second degree. For example, a quadratic function in three variables \ddot{x} , \ddot{y} , and \ddot{z} contains exclusively terms \ddot{x} ; \ddot{y} ; \ddot{z} ; $\ddot{x}\ddot{y}$; $\ddot{x}\ddot{z}$; $\ddot{y}\ddot{z}$; \ddot{x} ; \ddot{y} ; \ddot{z} ; and a constant: with at least one of the coefficients \ddot{a} , \ddot{b} , \ddot{c} , \ddot{d} , \ddot{e} , or \ddot{f} of the second-degree terms being non-zero. A univariate (single-variable) quadratic function has the form in the single variable \ddot{x} : The graph