STEM-GPT: Your Future AI Teaching Assistant?

Agatha Duzan | 310533 | agatha.duzan@epfl.ch
agatha-duzan

Abstract

In the age of digital education, it is crucial to improve the quality and accessibility of learning tools. This project introduces STEM-GPT, an AI assistant designed to answer multiple-choice questions on STEM subjects. Our aim is to enhance the learning experience of EPFL students by providing accurate and detailed explanations to their questions. To achieve that, we started from the base model GPT-Neo 125M and experimented improving it with various methods, such as Direct Preference Optimization and Supervised Fine-Tuning. Our final model achieves an overall accuracy of 39%, with a particularly strong performance in specific categories like biology where it achieves 72%.

1 Introduction

In the rapidly evolving landscape of digital education, leveraging advanced language models to enhance educational experiences presents a significant opportunity. With the increasing capabilities of AI, these tools can be used not only to provide correct answers but also to explain the reasoning behind them, enriching the learning process for students (Cacicio et al., 2023).

However, the highest-performing language models such as GPT-4 are not open source. This presents challenges in terms of data privacy, as professors and educational institutions may be reluctant to share their data with private companies (Sanderson, 2023). Additionally, these models are often computationally expensive, making them less accessible for individual use.

Our goal for this project is to create an open-source AI assistant that is efficient enough to run on personal devices, thus ensuring accessibility and privacy. Our final system, STEM-GPT, offers an interesting balance between size and capability.

To achieve this, we start with the model GPT-Neo 125M which provides a good trade-off between performance and computational requirements. We then enhance the model's capabilities through Direct Preference Optimization (DPO) and Supervised Fine-Tuning (SFT): DPO helps align the model outputs with human preferences, while SFT focuses on improving the accuracy on multiple-choice question answering.

This report details our methodology, the rationale behind our choices, the implications of our findings, and the ethical considerations associated with our work.

2 Related Work

This section covers previous work that has laid the foundation for our project. Our work builds upon advancements in large language models, direct preference optimization techniques, and evaluation methods for the task of multiple-choice question answering.

2.1 Large Language Models and GPT-Neo

In recent years, the development of large language models (LLMs) has significantly advanced natural language processing (NLP). One notable contribution is GPT-Neo (Black et al., 2021), a series of LLMs developed by Eleuther AI and trained on the extensive Pile dataset (Gao et al., 2020). The 125M parameter model, though the smallest in the series, excels in generating coherent and contextually relevant text by leveraging local attention mechanisms. Designed to replicate GPT-3-like capabilities, this model performs well in various NLP tasks and supports few-shot learning. Further evaluations (Kashyap et al., 2022) highlight its strengths and potential for fine-tuning in specific applications, making it an ideal candidate for our project.

2.2 Direct Preference Optimization

Direct Preference Optimization (DPO) is a technique designed to align language models with human preferences (Rafailov et al., 2023). Unlike tra-

ditional reinforcement learning from human feedback (RLHF), which involves complex and often unstable procedures, DPO uses a simpler approach by presenting preference pairs to guide the model toward desired behaviors. This technique is particularly relevant to our project, as it allows us to fine-tune the base model to produce more accurate and helpful responses for educational purposes.

2.3 Evaluation methods for MCQA

Evaluating the performance of language models on the task of multiple-choice question answering (MCQA) is crucial for assessing their effectiveness as educational tools. Several open-source datasets provide benchmarks for this evaluation. In this project, we used the Massive Multitask Language Understanding (MMLU) and AI2 Reasoning Challenge (ARC) datasets since they contain STEM-related questions.

The MMLU dataset (Hendrycks et al., 2020) offers a diverse collection of questions across various domains and difficulty levels, serving as a robust benchmark for measuring general knowledge and reasoning abilities of language models. Similarly, the AI2 Reasoning Challenge (ARC) dataset (Clark et al., 2018) focuses on science questions from elementary to high school levels, offering a diverse test bed for evaluating model performance on standardized academic content.

3 Approach

This section details our approach to develop an AI assistant capable of answering multiple-choice questions. Our methodology involves a sequence of steps starting from the base model, to applying DPO, to finally refining the model through either an extraction pipeline or fine-tuning to achieve the final system. The whole process is summarized in Figure 1.

3.1 Base model

The starting point is our base model: GPT-Neo 125M from EleutherAI, an autoregressive language model based on the transformer architecture. It consists of multiple layers of self-attention and feed-forward neural networks, designed to generate coherent and contextually relevant text. Its exact architecture is detailed in Table 1.

To adapt GPT-Neo for our specific application, we added a padding token to handle variable-length inputs efficiently.

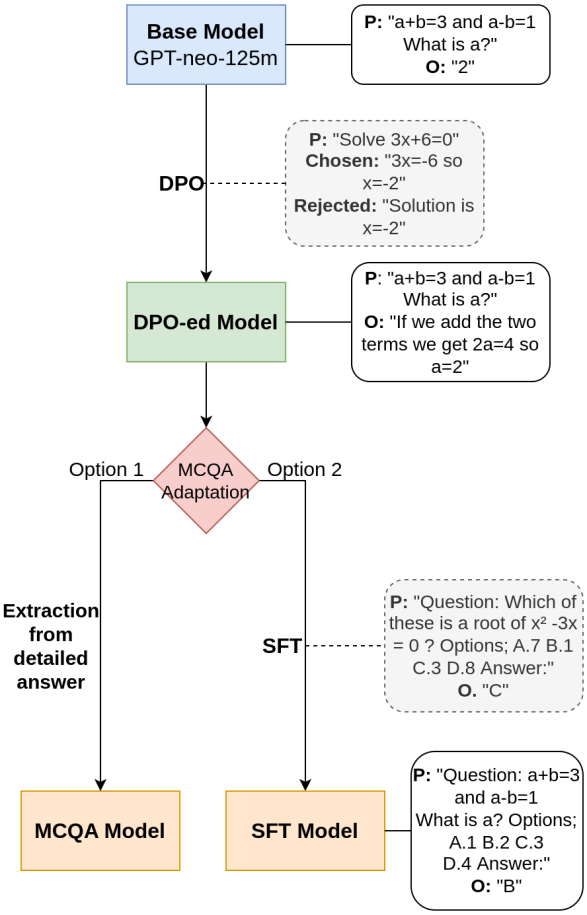

Figure 1: Base model to final models pipeline. In each example, "**P**" denotes the prompt and "**O**" the output.

Attribute	Value
Number of parameters	125 million
Hidden size	768
Number of layers	12
Number of attention heads	12
Vocabulary size	50257

Table 1: Base model architecture

3.2 Direct Preference Optimization

Since our task is answering STEM questions, we want our model to be helpful, correct in its answers, and overall aligned with the human judgement of what makes an answer 'good'.

That is why the next step of our process is aligning the base model's outputs with human preferences using DPO.

3.2.1 Preference data collection

As mentioned in subsection 2.2, we need a dataset of preference pairs to perform DPO.

During the first milestone of the project, each student received 100 prompts consisting of STEM

questions obtained from EPFL courses (with the teachers' consent). Using GPT-3.5, they generated two responses for each prompt: one 'better' and one 'less good,' thus forming a preference pair. All these preference pairs were then aggregated in a single dataset, ready to be used for DPO.

This data collection process provided us with a diverse set of human preferences that we aim to steer our model towards.

3.2.2 DPO training

With the collected preference data, we proceed to train the model using DPO: the core idea of this technique is to align the model outputs with human preferences by optimizing a policy objective.

More precisely, given a preference dataset $\mathcal{D}$ where each input x comes with a 'chosen' output y_c and a 'rejected' output y_r, we optimize the model policy π_θ compared to the reference policy π_{ref} so that it maximizes the difference in log-probabilities between chosen and rejected outputs. This process is given by the formula:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_c,y_r)\sim\mathcal{D}}\left[\Delta\right]$$

where Δ denotes the difference in log-probabilities between chosen and rejected outputs:

$$\Delta = \log\sigma\left(\beta\log\frac{\pi_\theta(y_c \mid x)}{\pi_{\text{ref}}(y_c \mid x)} - \beta\log\frac{\pi_\theta(y_r \mid x)}{\pi_{\text{ref}}(y_r \mid x)}\right)$$

β is a regularization parameter that controls the deviation from the reference policy π_{ref} to avoid overfitting.

3.3 Adapting to MCQA

At this stage, our model is capable of generating detailed answers to STEM-related questions. However, for multiple-choice questions (MCQ), we need to extract the specific letter corresponding to the choice made by the model. We explored two different methods to achieve this.

3.3.1 Detailed answer extraction

The first method, summarized in Figure 2, involves extracting a single letter answer from a detailed output. The steps are as follows:

1. Input the MCQ prompt and generate a detailed answer with our model
2. Append the phrase "Therefore, the correct answer is " to the generated answer
3. Identify the next token with the highest probability among the letters A, B, C, and D
4. Set this single letter as the final answer

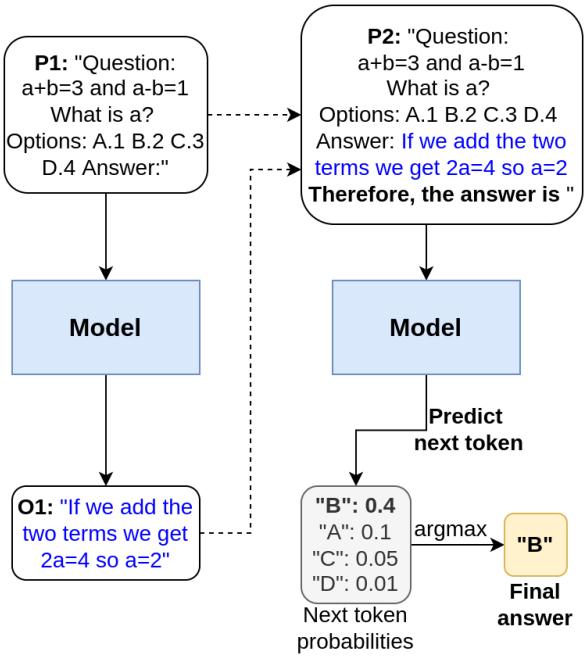

Figure 2: Extracting a single letter answer

3.3.2 Supervised Fine Tuning on MCQA

The second method involves supervised fine-tuning (SFT) our model on a dataset of MCQ questions and their corresponding single-letter answers.

During inference, we use the resulting model to generate only one token, which corresponds to one of the letters A, B, C, or D.

3.4 Final system

We will experiment with these different options and set our final system as the model with the best MCQA evaluation score.

4 Experiments

4.1 Data

In this subsection we describe the datasets used: from raw data, to preprocessing, to finally splitting for training and evaluation.

4.1.1 Preference data

The preference data was collected as explained in subsubsection 3.2.1. Each row in this dataset contains a preference pair with "prompt" being the question, "chosen" the better answer, and "rejected" the less good answer.

Preprocessing this data consisted of:

- removing rows where "chosen" and "rejected" were the same.
- removing rows where the prompt length or output length was above the 95th percentile.

- adding the system prompt: "*You are a helpful assistant for a STEM student*"

After preprocessing, the final DPO dataset contains 23,933 preference pairs, split into 21,539 pairs for training (90%) and 2,394 pairs for evaluation (10%).

4.1.2 MCQA data

For this project, we were provided with a MCQA dataset with 356 questions from machine learning, college biology, and college chemistry.

Additionally, we used open-source MCQA datasets as mentioned in subsection 2.3:

- MMLU dataset: we used only STEM-related categories, totaling 3,429 questions.
- ARC dataset: we discarded the 'Easy' category and used only the 'Challenge' category, as our final model is intended for university-level questions, resulting in 2,590 questions.

We had to preprocess this data to format all rows consistently with fields: "subject", "question" (formatted as '*Question: [our question] Options: A. [opt A] B. [opt B] C. [opt C] D. [opt D] Answer:*'), and "answer" (the single-letter correct answer).

In total, this gives us a final MCQA dataset of 6,375 questions split into 5,737 for training (90%) and 638 for evaluation (10%).

4.2 Evaluation

To assess the performance of our model, we focus on two main evaluation metrics: policy reward accuracy and MCQA accuracy.

4.2.1 Policy reward accuracy

To evaluate the impact of DPO, we assess the model's ability to distinguish between the 'chosen' and 'rejected' answers by looking at the scores assigned to each: a performant model should assign higher scores to the chosen answer.

The metric used here is accuracy, defined as the proportion of times the model correctly identifies the 'chosen' answer as better compared to the 'rejected' answer.

4.2.2 MCQA Accuracy

To assess the model's performance on MCQA, we evaluate the accuracy of the single-letter answers generated by the model, i.e. the proportion of questions for which the model's predicted answer matches the correct answer.

4.3 Baselines

For our experiments, we compare the performance of our model to two baselines: the base model GPT-Neo-125M, and the model GPT-2-124M by OpenAI (Radford et al., 2019) whose architecture is very similar. These comparisons help quantify the improvements achieved through our methods.

It is also important to note that random choice would yield around 50% policy reward accuracy (distinguishing between 'chosen' and 'rejected') and 25% accuracy in MCQA (choosing between A, B, C, and D).

4.4 Experimental details

The hyperparameters used in our experiments are shown in Table 2.

- The regularization β for DPO was chosen based on the value reported in the original paper (Rafailov et al., 2023)
- The learning rates were explored within a range seen in literature (Kashyap et al., 2022), and the best learning rates for both DPO and SFT were determined through experimentation
- The number of epochs was set to 3, as no significant progress was observed beyond this point (performance plateauing)
- We used AdamW the default optimizer from the transformers library, as it improves generalization and training stability by decoupling weight decay from gradient updates

Hyperparameter	Value(s)
Optimizer	AdamW
Learning rate	$\{5 \cdot 10^{-i}\}_{i=5}^{7}$
Best learning rate (DPO)	$5 \cdot 10^{-7}$
Best learning rate (SFT)	$5 \cdot 10^{-5}$
Number of epochs	[2, 5]
Best number of epochs	3
DPO regularization (β)	0.1

Table 2: Hyperparameter values used in the training setups

The runtimes for the various steps of our project are detailed in Table 3. All tasks were executed on a single GPU with 64GB RAM.

Task	Runtime
DPO training	2 hours
DPO evaluation	5 minutes
MCQA evaluation (by answer extraction)	45 minutes
MCQA finetuning	5 minutes
MCQA evaluation (on finetuned model)	10 seconds

Table 3: Runtimes for different steps of the project

4.5 Results

4.5.1 Policy reward accuracy

The results in Table 4 show that applying DPO significantly improves the model's ability to identify the better answer, with the DPO-ed model achieving a reward accuracy of 56% compared to the baselines 24%.

	Model	Reward Accuracy
Baselines	GPT-2	24.0%
	GPT-Neo	24.2%
With DPO	**GPT-Neo + DPO**	**56.1%**

Table 4: Policy reward accuracy before vs after DPO

4.5.2 MCQA accuracy

The results in Table 5 show the different model accuracies on MCQA using the two adaptation methods. The comparison includes an ablation study to evaluate the impact of DPO on MCQA accuracy.

	Model	MCQA Accuracy
With Extraction	GPT-2	23.5%
	GPT-Neo	22.7%
	GPT-Neo + DPO	**24.1%**
With MCQA SFT	GPT-2 + SFT	26.1%
	GPT-Neo + SFT	**39.3%**
	GPT-Neo + DPO + SFT	33.1%

Table 5: Comparison of MCQA accuracy for various MCQA adaptation methods

Among the models using the extraction method, we see that DPO yields a marginal improvement compared to the base model. However, all accuracies are close to the random choice baseline of 25%. One possible explanation is that when the model generates a detailed answer, it produces a lot of additional information that is not directly related to the single-letter answer choice: this verbosity can introduce noise and distract the model when selecting the correct letter in the end.

On the contrary, the SFT method shows significant improvement in MCQA accuracy for GPT-Neo, with GPT-Neo + SFT achieving 39%. In this case however, the combination of DPO and SFT results in a lower accuracy of 33%.

Given these results, our final system STEM-GPT will be based on the GPT-Neo model fine-tuned with the SFT method, as it provides the best performance in MCQA evaluation.

5 Analysis

This section includes further analysis on the comparative performance of different models, the impact of DPO, and a detailed evaluation of our final system's performance across different subjects.

5.1 Comparison of GPT-Neo and GPT-2 Performance

An interesting result seen in Table 5 is that although the baseline models GPT-Neo-125M and GPT-2-124M have very similar architectures and were SFT-ed on the same data in our experiments, GPT-2 + SFT achieves an accuracy close to the random choice baseline (26%) while GPT-Neo + SFT achieves the best accuracy overall (39%).

One possible explanation behind this drastic difference is that GPT-Neo was trained on the Pile dataset (Gao et al., 2020) which has significantly more data (825 GB) than WebText (Radford et al., 2019), the dataset on which GPT-2 was trained (40 GB).

5.2 Ablation study

The ablation study investigates the impact of DPO on the final model's performance on MCQA.

When using the extraction method, DPO slightly improved the base model's performance, with the DPO-ed model achieving 24% accuracy compared to 23% for the base model. This indicates that the DPO-ed model's detailed answers were more 'useful', resulting in marginally higher accuracy in MCQA.

However, for the SFT method, the fine-tuned base model achieved better performance (39%) than the fine-tuned DPO-ed model (33%). This might imply that fine-tuning on single-letter answering did not leverage the step-by-step reasoning learned during DPO effectively.

5.3 STEM-GPT detailed evaluation

To further understand the performance of our final system, we evaluated its MCQA accuracy on subsets of our data, categorized by subject. The results are illustrated in Figure 3.

We see that the model performs best on subjects like biology and chemistry, while performing

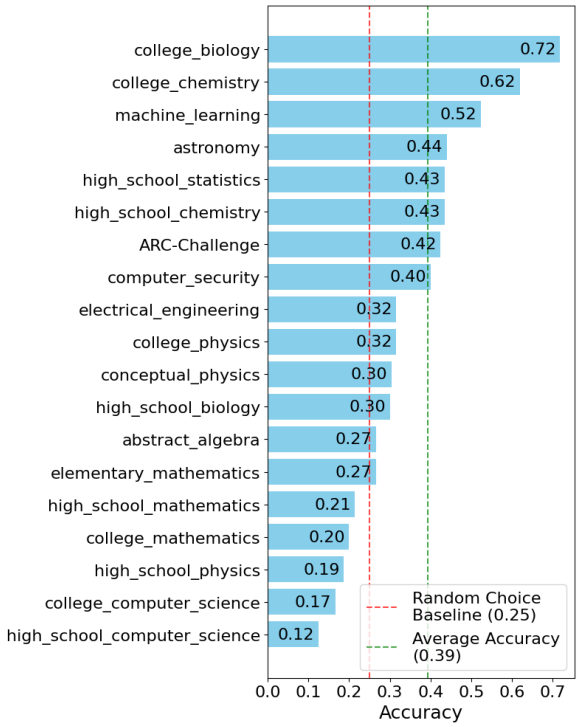

Figure 3: Detailed evaluation of our best final system: MCQA accuracy by question category

worst on subjects like computer science, math, and physics.

One possible explanation is that subjects like biology and chemistry often involve descriptive and explanatory content, which aligns well with a LLM's strength in processing and generating natural language.

On the contrary, fields like computer science, mathematics, and physics frequently involve symbolic manipulation, formal proofs, and problem-solving techniques that may not be as effectively handled by a LLM trained primarily on natural language. The complexity and abstract nature of these subjects might pose additional challenges.

6 Ethical considerations

This section explores the broader ethical implications of our work, including the potential for other languages adaptation, benefits/harms balance, and the potential impact on vulnerable or marginalized groups.

6.1 Adaptation to other languages

Adapting our model to other languages is crucial to ensure global accessibility, inclusivity, and educational equity.

For high-resource languages like French and German, this could be achieved by fine-tuning on large, high-quality datasets available for these languages (like Wikipedia Dumps or ArXiv).

For low-resource languages such as Urdu and Swahili, a different approach would be needed. We could use techniques such as transfer learning, where the model is first trained on a high-resource language and then fine-tuned on the low-resource language data.

6.2 Interaction with signed language users

Adapting the model to interact with users in signed languages (SL) is a more complex task, as it involves multimodal capabilities.

One approach could be to develop a multimodal AI assistant that integrates NLP with computer vision. This assistant could interpret SL through video input, translate it into text, and respond appropriately.

It would be crucial to collaborate with experts and incorporate datasets specifically for SL.

6.3 Potential benefits and harms

If our model works as intended, it could significantly benefit students and educators by providing accurate and detailed explanations for STEM-related questions. This could help democratize the access to educational resources by making advanced learning tools available to a broader audience.

However, there are potential harms to consider. Our model could increase misinformation by generating incorrect or misleading information, which is especially concerning in an educational context.

Since the base model was primarily trained on English data, there could also be disparities in performance when adapted to other languages: this could lead to less effective learning tools for non-English speakers.

To mitigate these risks, robust evaluation and monitoring mechanisms should be implemented to ensure the accuracy and fairness of the model outputs. We should also be transparent about the model limitations and set guidelines for a responsible use.

6.4 Potential impact on vulnerable or marginalized groups

Not only could our model have lower performance in other languages, but it could also be more prone to harmful behavior in these languages. This includes the propagation of biases present in the train-

ing data, which may disproportionately affect vulnerable or marginalized groups. Such biases could lead to unfair or even discriminatory outputs.

To minimize these risks, it would be essential to carefully curate and balance the training datasets, and evaluate our model adversarially across different groups and languages.

7 Conclusion

In this project, we created STEM-GPT: an AI assistant for answering STEM-related multiple-choice questions, based on the model GPT-Neo 125M.

We explored the effectiveness of different methods to enhance its performance, and found that SFT alone yielded the best accuracy, outperforming other method combinations.

Detailed evaluation showed that STEM-GPT performs well in more descriptive subjects like biology and chemistry but struggles with more abstract subjects like computer science, math, and physics.

A future improvement could be to develop a multi-modal AI assistant that can interpret text and visual information such as graphs and figures, which are crucial in STEM education.

Future work could also focus on crafting a more advanced and diverse STEM dataset, adding data in multiple languages, and possibly using AI to generate data.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Sarah Cacicio, Adult Literacy, and Learning Impact Network. 2023. Chatgpt: Leveraging ai to support personalized teaching and learning. *Adult Literacy Education*, page 70.

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*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Rohan Kashyap, Vivek Kashyap, et al. 2022. Gpt-neo for commonsense reasoning–a theoretical and practical lens. *arXiv preprint arXiv:2211.15593*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Katharine Sanderson. 2023. Gpt-4 is here: what scientists think. *Nature*, 615(7954):773.

A Appendix

A.1 AI Usage

I wanted to use ChatGPT to help me with debugging code related to the trl library (since each function has a million optional arguments) but it kept hallucinating arguments even when provided with the source code of a class (ex: DPOTrainer), so in the end I just spent hours reading the documentation myself.

This was pretty fastidious and I was definitely disappointed at first that the AI shortcut didn't work as expected, but I definitely learned a lot in the end.

On the contrary, ChatGPT was super helpful with LaTeX formatting, notably to convert screenshots of dataframes to clean LaTeX tables (which could have been done manually but I was curious to see the accuracy).