The EPFL Study Companion: AI-Powered Academic Assistance

Kelyan Hangard | 312936 | kelyan.hangard@epfl.ch
César Camus-Emschwiller | 328075 | cesar.camus-emschwiller@epfl.ch
Jennifer Abou-Najm | 312656 | jennifer.abou-najm@epfl.ch
NLPain

Abstract

The goal of this project is to develop a specialized Large Language Model tailored to EPFL course material, serving as an AI tutor to provide detailed, helpful responses without human interaction. Using T5-small as the base model, we enhance its scientific question-answering ability through supervised fine-tuning, align its outputs with human preferences via Direct Preference Optimization, and finally, fine-tune it again to generate single-letter responses for multiple-choice questions. After extensive experimentation and hyper-tuning, the final model obtained presents an MCQA accuracy of 32% on an EPFL-based testing dataset, outperforming models such as T5-base, which has three times more parameters, demonstrating the effectiveness of our approach. Moreover, advanced quantization techniques such as LLM.int8() were explored, leading to the final quantized model achieving an accuracy of 29.99%, while reducing the model size by more than 59%.

1 Introduction

EPFL welcomes over 2,000 new bachelor students annually. Typically, for every 10 first-year students, one student assistant is needed for each course, which usually amounts to around 5 courses per semester. This creates a demand for approximately 1,000 student assistants per semester. Managing such a large number of assistants requires substantial resources, and despite these efforts, students often face long wait times to get their questions answered. This delay can hinder their understanding of course material, which is crucial for academic success in such a challenging environment.

To address this challenge, we propose developing a specialized Large Language Model (LLM) tailored to EPFL course material. This model aims to function as an AI tutor, providing students with detailed and helpful responses to their inquiries without requiring human interaction. By leveraging this AI tutor, we can significantly reduce the need for student assistants and enable students to obtain answers to their questions more quickly, at any time, and without the wait associated with crowded exercise sessions. Additionally, to enhance accessibility for small interfaces such as phones, we decided to prioritize smaller model configurations.

Building on the T5-small pre-trained seq2seq model (Raffel et al., 2023), we first perform supervised fine-tuning (SFT) to teach it how to respond to scientific questions in a helpful manner. We then apply Direct Preference Optimization (DPO), as proposed in (Rafailov et al., 2023a), to align the model's responses with human preferences, helping it understand what constitutes a good answer versus a bad one. Finally, we fine-tune the model again to provide concise, single-letter responses using chain-of-thought (COT) techniques for multiple-choice questions (MCQA).

Our experimental findings reveal promising outcomes, given the relatively small number of parameters (60M), with the final model achieving a 32% accuracy on a selected test dataset derived from EPFL course materials. This performance surpasses random guessing and considered baselines.

Finally, we explore various quantization methods, including advanced techniques like LLM.int8(), designed to handle outlier challenges and achieve substantial size reduction with minimal performance degradation. Through rigorous experimentation, our quantized model achieves a reduced size of 98MB from an initial 242MB, while maintaining an accuracy of 29.99%.

2 Related Work

Transfer learning has revolutionized NLP by enabling models to leverage knowledge from large-scale datasets and apply it to specific tasks. Notable works include the T5 model by (Raffel et al., 2023), which demonstrates the effectiveness of a text-to-text transfer learning framework, and BERT by (Devlin et al., 2019), which introduced a powerful bidirectional encoder for language understanding. Our approach aligns with these methods by utilizing the T5-small pre-trained seq2seq model, chosen for its balance of performance and computational efficiency. Unlike these general-purpose models, our work specifically tailors the pre-trained model to handle EPFL course ma-

terial, making it uniquely suited for educational purposes.

Fine-tuning pre-trained models on domain-specific data is crucial for achieving high performance in specialized tasks. Techniques such as Supervised Fine-Tuning (SFT) have been successfully employed to adapt models like T5 and BERT for specific applications, including scientific question answering (Raffel et al., 2023) and computer science theory QA (Mateen, 2023a). In our project, we mirror these strategies by fine-tuning the model on EPFL course materials, thereby enhancing its ability to provide detailed and contextually accurate responses to academic questions.

Aligning model outputs with human preferences enhances the quality and relevance of generated responses. (Rafailov et al., 2023b) introduced Direct Preference Optimization (DPO), a method that refines model responses based on human feedback, ensuring that the answers are not only correct but also aligned with user expectations. Our implementation of DPO further refines the model's alignment with user preferences, thereby improving the overall user experience for students interacting with the AI tutor.

Effective evaluation of NLP models is essential for continuous improvement. Metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) have been widely adopted for evaluating the quality of generated text. Additionally, BERTScore (Zhang et al., 2019) leverages contextual embeddings from BERT to provide a more nuanced assessment of text similarity. Our project adopts these evaluation metrics to assess the performance of our AI tutor, ensuring that the model meets high standards of accuracy and coherence in its responses.

The availability of large-scale, high-quality datasets is a cornerstone of successful NLP research. The English Colossal Clean Crawled Corpus (Dodge et al., 2021a) and the Stack Exchange Question Pairs dataset (Team, 2021b) are examples of datasets that have significantly contributed to training robust language models. Our project leverages a dataset composed of EPFL course materials, which provides the necessary domain-specific data to fine-tune our model effectively.

3 Approach

3.1 System Architecture

The architecture of our system combines the robustness of a pre-trained language model with advanced fine-tuning techniques, incorporating both supervised learning and direct preference optimization.

3.1.1 Base Model Selection

The foundation of our AI tutor is the T5-small pre-trained language model (Raffel et al., 2023), which consists of 60M parameters and has been pre-trained on a mixture of unsupervised and supervised tasks from the Colossal Clean Crawled Corpus (Dodge et al., 2021b). This seq2seq model with encoder-decoder architecture, was selected for its relatively small size, simplifying the training process and increasing accessibility for smaller interfaces, for its robust performance across various NLP tasks, and extensive pre-training on diverse datasets, ensuring a comprehensive understanding of language, essential for adapting to specialized domains such as EPFL course material.

3.1.2 Supervised Fine-Tuning for Scientific Question Answering

To teach the model how to answer scientific questions in a detailed and helpful manner, SFT is employed. This method optimizes the following objective function, designed to maximize the likelihood of the model predicting the correct answers based on the given questions:

$$\mathcal{L}_{\text{SFT}} = -\sum_{(x_i,y_i)\in\mathcal{D}} \log P(y_i|x_i;\theta) \tag{1}$$

where $\mathcal{D}$ represents the dataset of input-output pairs (x_i, y_i), and θ denotes the model parameters.

3.1.3 Alignment through Direct Preference Optimization

Following the initial fine-tuning process, we further refine the model's responses to align them closely with human preferences with DPO (Rafailov et al., 2023a). The following loss function is used, aiming to minimize the discrepancy between the model's predicted responses and the preferred responses provided by human evaluators:

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

$$\text{where} \quad f = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}$$

Here, π_θ represents the optimal policy, π_{ref} the reference policy, y_w the "winning" response and y_l the "losing" response given the input x.

3.1.4 Supervised Fine-tuning for MCQA

The final stage involves fine-tuning the model again to transition its output format from detailed step-by-step explanations to succinct, single-letter responses suitable for multiple-choice questions. This last SFT step ensures that the model can efficiently handle MCQA scenarios,

which is the assessment format for this project. The output format of the model here is structured as "[*detailed explanation*] + The correct answer is: [*letter*]", enabling straightforward extraction of the letter using a simple regex.

3.2 Dataset Creation

Since the goal is to train an LLM specialized in the EPFL course material, it is crucial to obtain a dataset tailored specifically for that purpose. To create these samples, MNLP students received a set of EPFL course questions and used a given ChatGPT wrapper to generate preference pairs.

Each pair consists of a high-quality answer and a medium-quality answer, including information about correctness, clarity, completeness, conciseness, and other relevant factors. For collecting the high-quality answers, our prompting method involved several strategies to elicit detailed and accurate responses. This included explicitly mentioning the AI's role by stating "You are a teaching assistant in [*specific domain*]", using only positive language, providing encouragements such as "Think step by step" and "Be clear", and asking for clarifications when needed. Additionally, we offered extra hints if the model struggled. Conversely, for the medium-quality answers in the pairs, the AI role was provided, but without encouragements or additional hints. This ensured a varied dataset that could help the model distinguish between different levels of response quality.

3.3 Specialization: Quantization

The objective here is to achieve the maximum reduction in size of the final model with minimal performance loss. To achieve this goal, we employed a combination of advanced quantization techniques. First, as introduced in (Dettmers et al., 2022), we address the performance degradation caused by outlier features in matrix multiplication computations using the LLM.int8() approach, which operates as follows:

1. Identify outliers from the input hidden states, specifically values exceeding a predefined threshold, categorized by columns.

2. Conduct matrix multiplication of the outliers using FP16 precision and the non-outliers using int8 precision.

3. Dequantize the results obtained from non-outliers and integrate them with the outlier results, achieving the final result in FP16.

Additionally, the model is loaded initially using 4-bit precision, employing NF4 data types for computation with bfloat16. Furthermore, we apply double quantization, which involves further quantizing the constants obtained from the initial quantization process.

4 Experiments

4.1 Data

4.1.1 Data Sources for Initial SFT

First, to refine the model's understanding of what constitutes a high-quality response, we rely on the following question-answering datasets for SFT:

- ScienceQA (Lu et al., 2022): MCQs dataset with a diverse set of science topics and annotations of their answers with corresponding lectures and explanations. By filtering out MCQs that require image context and selecting only those within the Physics and Engineering categories aimed at grade levels above 6, we identify 265 instances.

- MathInstruct (Xiang Yue, 2023): Instruction-tuning dataset compiled from 13 math rationale datasets. We specifically choose a subset of 1445 instances, derived from the aqua-rat-filtered source, as it consists of MCQs that require writing Python programs.

- CS-Theory-QA (Mateen, 2023b): Comprehensive dataset comprising 172 theoretical computer science questions across multiple domains like operating systems and machine learning.

- Preference Pairs from MNLP students in SFT format: In order to align the model with EPFL question types before performing DPO, we select at random one of the preferred options for each question and obtain 1445 instances.

4.1.2 Data Sources for DPO

In order to perform DPO, we will use the following preference pair datasets:

- Preference Pairs from MNLP students: Dataset containing preference pairs of answers from given ChatGPT wrapper to EPFL questions. 2961 instances are selected.

- Stack Overflow (Team, 2021a): Dataset consisting of the most-voted (best) and less-voted (worse) answers for questions asked on the Stack Exchange forums. Selecting only relevant domains from mathematics to computer science, we obtain 2961 instances.

4.1.3 Data Sources for Final SFT

As a final step, to transition the answer format from detailed explanations to single-letter responses, we use the following datasets (also utilized in the initial SFT stage), and filtered to only contain MCQAs: Preference Pairs from MNLP students in SFT format and ScienceQA. However, the output format here is: "[*detailed explanation*] + The correct answer is: [*letter*]". To obtain the corresponding correct letters, we use the given ChatGPT wrapper, providing it with the question and detailed answer and instructing it to output the letter corresponding to the given answer.

4.1.4 Dataset Processing and Splits

The datasets are filtered to include only instances with lengths smaller than the maximum input token length for prompts, chosen, and rejected answers (512 tokens for the seq2seq models and 1024 tokens for the causal models). Additionally, the datasets are divided into training, validation, and testing sets, ensuring that parts of the SFT and DPO datasets sourced from the same dataset (MNLP preference pairs) are appropriately managed. For instance, special care is taken to ensure that the test sets do not include questions that the model was trained on.

4.1.5 Legal and Ethical Data Considerations

The data used in this project is sourced from publicly available datasets and contributions from students, ensuring compliance with relevant data usage policies. Ethical considerations include ensuring the privacy and consent of contributors and maintaining the integrity of the data to avoid biases in model training.

4.2 Evaluation Methods

The quality of the answers generated by the models is evaluated using ROUGE (Lin, 2004) [R1, R2] and BLEU (Papineni et al., 2002) [BL] metrics, which are based on n-gram overlap, as well as BERTScore (Zhang et al., 2019) [BS], which captures semantic similarity and is more robust to phrasing variations. Since the goal is to produce high-quality answers to EPFL questions, the test set comprises questions from the given MNLP preference pairs, with one preferred answer chosen at random. To ensure the model generalizes well, we also evaluate it on the MMLU benchmark (Hendrycks et al., 2021), selecting only STEM categories.

To evaluate the alignment to human preference, the models' reward accuracy [ReAcc] is computed on a test set sampled from the MNLP preference pairs.

Moreover, to assess the model's accuracy and correctness, we compute the accuracy [Acc] on a test set composed of MCQs sampled from the MNLP preference pairs, where the correct answer, in letter format, corresponds to the preferred answer.

In addition to quantitative metrics, the quality of the model's generated text will be analyzed through qualitative human evaluation, to ensure the coherence, relevance, and overall quality of the responses produced by the model.

It should be noted that the first two evaluation methods (R1, R2, BL, BS, ReAcc and qualitative evaluation) are performed on the model before the final SFT stage, while the last evaluation (Acc) is conducted after it.

4.3 Baselines

We have chosen the pre-trained model T5-small as a baseline, along with two other similar models to ensure a meaningful comparison: FLAN-T5-small (Chung et al., 2022), which has been fine-tuned on more tasks, and T5-base, which has 220M parameters. To compare their MCQA accuracy, these models were also fine-tuned on the second-stage SFT training dataset, however, with the output format "[*letter*]".

4.4 Results

4.4.1 Choice of Base Model

The first step involves determining which base model is most appropriate for our task. Prioritizing smaller models to simplify the training and generation processes, we experimented with three potential base models: the T5-small model (Raffel et al., 2023), consisting of 60M parameters and pre-trained on a mixture of unsupervised and supervised tasks from the Colossal Clean Crawled Corpus (Dodge et al., 2021b); the FLAN-T5-small model (Chung et al., 2022), with 80M parameters and fine-tuned on a broader range of tasks; and the GPT-2 small model (Radford et al., 2019), with 124M parameters, pre-trained on the WebText, composed of 8M webpages. The first two models are seq2seq models with an encoder-decoder architecture, while the latter is a causal language model with a decoder-only architecture.

After some experiments, the hyperparameters are set to reasonable values: a learning rate of 0.001 with the Adafactor optimizer on 1 epoch for SFT, a beta of 0.1 on 1 epoch for DPO (other settings are set to default). SFT training is conducted on 1959 instances, while DPO training on 4737 instances. Evaluation is performed on our question-answering test dataset composed of 77 instances, and reward accuracy is assessed on a subset of 200 instances from our testing preference pairs dataset to speed up evaluation.

As shown in Table 1, the T5-small model demonstrates the best performance in terms of BERTScore and reward

Model	Type	R1	R2	BL	BS	ReAcc
T5	Base	0.198	0.106	0.001	0.817	0.305
T5	SFT	0.227	0.106	0.024	0.819	0.335
T5	DPO	0.261	0.131	0.010	**0.831**	**0.64**
FLAN-T5	Base	0.037	0.012	0.000	0.774	0.285
FLAN-T5	SFT	0.193	0.072	0.012	0.811	0.325
FLAN-T5	DPO	0.165	0.062	0.012	0.791	0.335
GPT-2	Base	0.270	**0.178**	0.047	0.808	0.315
GPT-2	SFT	0.259	0.123	0.052	0.817	0.325
GPT-2	DPO	**0.313**	0.148	**0.071**	0.822	0.61

Table 1: Performance comparison of mdels

Parameter	Value	R1	R2	BL	BS	ReAcc
default	-	0.261	0.131	0.010	0.831	0.64
$n_{epochs-SFT}$	2	0.286	0.155	0.011	0.840	0.592
$n_{epochs-DPO}$	2	0.276	0.137	0.036	0.822	0.612
lr	10^{-2}	0.006	0.000	0.000	0.718	0.64
lr	10^{-4}	0.287	0.136	0.024	0.832	0.64
β	0.05	0.292	0.134	0.048	0.821	0.642
β	0.2	0.291	0.127	0.055	0.821	0.625

Table 2: Hyperparameter tuning results

accuracy, which are our most robust metrics. Based on these results, we decide to proceed with the T5-small model.

4.4.2 Hyperparameters Tuning

The next step is to determine the optimal hyperparameters for this model. Table 2 summarizes the performances after DPO, varying the learning rate (lr), beta parameter (β), and number of epochs (n_{epochs}), and keeping all other parameters to previous values.

First, regarding the number of epochs, the best reward accuracy is observed with one epoch for both SFT and DPO (although using two epochs for SFT improves other metrics, accuracy is prioritized in this context). Second, the best performance in terms of generation metrics are obtained with a learning rate of 10^{-4}. It is interesting to note that the accuracy remains the same for all three learning rates considered. Third, regarding the parameter β, decreasing its value seems to increase ROUGE scores and accuracy. Indeed, a better performance in terms of ROUGE, BLEU and reward accuracy are obtained with a β of 0.05.

Taking all that into account, and after further experimentation, the final hyperparameters used are: lr of 10^{-4}, β of 0.09 and 1 epoch of SFT and DPO. The model thus obtained presents a reward accuracy of 0.652 on the test dataset subset that was previously used, and of 0.605 on the complete test dataset, composed of 1136 preference pairs. Reports of other metrics, after second-stage SFT, can be found in Table 5.

Model	Acc-Test	F1-Test	Acc-MMLU	F1-MMLU
final model	0.320	0.254	0.314	0.270
T5-s	0.271	0.117	0.219	0.079
FLAN-T5-s	0.266	0.117	0.227	0.131
T5-b	0.293	0.222	0.273	0.209

Table 3: Final model MCQA accuracy comparison

4.4.3 Further Experimentation

To explore potential improvements, other types of model architecture were tested, such as applying DPO a second time after the second-stage SFT, using the same output format as the second-stage SFT (explanation followed by the correct answer). Furthermore, applying this format on initial-stage SFT and DPO was also tested. However, the MCQA accuracies obtained for these methods on the testing dataset, which comprised 646 instances, were around 25%, which is relatively lower compared to the chosen model architecture.

4.4.4 Final Model Performance Comparison

Table 3 presents the computed metrics (MCQA accuracy [Acc], F1-score [F1] on the MCQA testing dataset [Test] composed of 646 instances, and MMLU benchmark [MMLU], composed of 1251 instances, for our final developed model, using the previously mentioned hyperparameters and specified training datasets. These results are compared to the metrics for our three chosen baselines: T5-small [T5-s], FLAN-T5-small [FLAN-T5-s] and T5-base [T5-b].

We can thus see that our final model consistently presents the best performance. Moreover, it is interesting to note that it even outperforms T5-base, which has a significantly higher number of parameters. Focusing on the results of our model, it achieves an accuracy of 32% on the testing set. While this may not seem very high, it is notably better than random guessing. This is particularly impressive given the model's relatively low parameter count and the high difficulty level of EPFL questions.

4.4.5 Quantization

The primary objective here is to achieve the maximum reduction in model size while minimizing performance loss in the final developed model. Our experiments involved various techniques, including double quantization, selecting the appropriate bit precision for model loading, exploring different quantization schemes data types, and determining the optimal data type for computations. Most relevant experiments and corresponding results are outlined in Table 4, where [model-load] represents the bit precision in which the model is loaded, [double-quant] determines if double quantization is applied or not, [int8-

Option	1	2	3	4	original
model-load	4 bit	4 bit	4 bit	8 bit	-
double-quant	Yes	Yes	No	No	-
int8-thresh	1	6	6	6	-
comp-prec	bf16	bf16	bf16	f32	-
R1	0.275	0.279	0.278	0.273	0.363
R2	0.134	0.134	0.134	0.136	0.184
BL	0.02	0.020	0.021	0.022	0.117
BS	0.831	0.833	0.832	0.834	0.852
ReAcc	0.509	0.602	0.602	0.610	0.652
Acc	0.298	0.299	0.3	0.301	0.32
Size [MB]	98.9	98.9	107	114	242

Table 4: Quantization experimental results

Model	Size	R1	R2	BL	BS	ReAcc	Acc
final	242MB	0.430	0.280	0.155	0.874	0.605	0.32
quant	98MB	0.428	0.275	0.158	0.871	0.598	0.299

Table 5: Comparison of final (quantized) models

thresh] corresponds to the threshold for the LLM.int8() approach and [comp-prec] the computational type. Please note that the results are reported for the MNLP question-answering and MCQA testing datasets previously mentioned.

As option 2 leads to the best performance over size reduction ratio, we will proceed with it. The final results with this method, compared to the final model, can be found in Table 5. Please note that these results are computed after the second-stage SFT, while previous results in Table 4 are computed before the second-stage SFT, except for the accuracy. It is noticeable that there is a significant increase in certain performance metrics, which can be explained by the fact that the model's output predictions include "The correct answer is:", ensuring a match with specific parts of the gold answers.

5 Analysis

5.1 Qualitative Evaluation

The generations of the final model were also qualitatively evaluated to assess their quality and coherence. Generally, the answers generated are sensible and adhere to the required format. However, the model seems to struggle with complex COT reasoning, which is expected given the relatively small size of the chosen model.

For example, consider the question: "Which statement about Singular Value Decomposition (SVD) is true? Options: **A.** The original matrix M can be reconstructed from the matrices K and D computed by the SVD **B.** SVD, after eliminating the largest singular values, can be interpreted as a least square approximation to the original matrix **C.** The singular values can be computed from the Eigenvalues of the original matrix M **D.** SVD can be computed by constructing Eigenvectors of matrices derived from the original matrix M". The model's answer is: "Singular Value Decomposition (SVD) can be reconstructed from the matrices K and D computed by the SVD. Singular Value Decomposition (SVD) can be interpreted as a least square approximation to the original matrix. The correct answer is: A". Here, while the chosen answer is technically correct, the explanation actually corresponds to another option and is incorrect.

Another example is: "Tick the *incorrect* assertion. The security of the DH protocol requires . . . ? Options: **A.** that the DL problem is hard. **B.** that the factoring problem is hard. **C.** that we use groups of prime order. **D.** a KDF function." The model's response was: "The incorrect assertion is: that the DL problem is hard. Explanation: The security of the DH protocol requires that the DL problem is hard. The correct answer is: B". Here, the model's justification is trivial and lacks depth, simply restating the assertion rather than providing a detailed explanation. Moreover, the letter chosen by the model does not correspond to its first prediction. This might be due to the imbalance in the training data, which consisted of approximately 32% of answers corresponding to option B.

In both examples, the model provides contextually relevant answers in the correct format; however, they lack comprehensive reasoning and detail, highlighting a limitation in its ability to handle more complex, nuanced explanations. This limitation is likely due to the smaller parameter size of the model, suggesting that a larger model might perform better in generating more detailed and contextually rich explanations.

5.2 Ablation study

To understand the impact of each of the first two stages of our model development, we conducted experiments by omitting each stage individually. Specifically, we ran one experiment without the initial SFT stage, one without the DPO stage, and one without both, using the corresponding training datasets and hyperparameters previously specified. The third stage, which ensures the model outputs single-letter responses for MCQA accuracy computations, was mandatory and included in all experiments. The results of these experiments on the MNLP questions answering and MCQA testing dataset previously specified, are summarized in Table 6.

For all cases, better results are attained using our full approach, demonstrating its effectiveness. However, it is interesting to note that removing the first SFT stage does not significantly influence the results, which could be explained by the presence of the second-stage SFT,

Stage removed	R1	R2	BL	BS	ReAcc	Acc
final	0.430	0.280	0.155	0.874	0.605	0.32
SFT	0.415	0.230	0.1	0.869	0.595	0.279
DPO	0.427	0.245	0.139	0.871	0.325	0.308
both	0.407	0.215	0.119	0.831	0.305	0.252

Table 6: Ablation study performance metrics results

that might be introducing a similar COT effect. Moreover, while removing the DPO stage significantly decreases reward accuracy, as expected, it can be noticed that the MCQA accuracy is not greatly affected.

6 Ethical Considerations

6.1 Adaptation to Other Languages

For adapting the model to handle different languages, switching to the multilingual version of T5, mT5 (Xue et al., 2021), pre-trained on the mC4 corpus which covers 101 languages, could be beneficial. Alternatively, training multilingual adapters between the layers of the pre-trained network, or incorporating a translation mechanism before processing the input could also facilitate adaptation to other languages.

6.1.1 High-Resource Languages

For high-resource languages like French and German, there is typically a wealth of available data, including EPFL first-year content that is usually available in these two specific languages. The adaptation process would involve fine-tuning the model using large, high-quality datasets in the target language, such as news articles, educational content, and scientific literature. Given the extensive availability of these resources, the model can be trained effectively to understand and generate responses in these languages with a high degree of accuracy.

6.1.2 Low-Resource Languages

Adapting the model to handle low-resource languages poses unique challenges due to the lack of data. The first step would involve gathering all accessible text data in the target language, which could include news articles, translations of scientific literature, etc. To augment the dataset, techniques such as back-translation, where text is translated to another language and back, could be employed to increase the volume and diversity of training data.

Moreover, human feedback is crucial in such cases. Engaging native speakers to generate and evaluate preference pairs could ensure that the model's answers are aligned with cultural and contextual nuances. Continuous evaluation of the model with native speakers' input, using both quantitative metrics and qualitative feedback would also improve the model's performance in understanding and generating the low-resource language. This approach would help in refining the model to be both linguistically and contextually relevant.

6.2 Adaptation to Signed Languages

6.2.1 Model Architecture

The model architecture needs to be adapted to handle video inputs rather than text. This may involve integrating convolutional neural networks or other video processing techniques to extract features from sign language videos. Alternatively, transformer models designed for video processing, like the Vision Transformer (Dosovitskiy et al., 2020), could be employed to handle the temporal and spatial dimensions of sign language.

6.2.2 Interactive Interface

It would be useful to develop an interactive user interface that can capture real-time video input from users signing and provide immediate feedback, by displaying generated sign language through an animated avatar or synthesized video to ensure clarity and accuracy in communication.

6.2.3 Sign Language Recognition and Generation

For sign language recognition, the model must be able to interpret the visual input and translate it into a semantic representation. This involves detecting and recognizing individual signs as well as understanding the sequence and context in which they are used.

For sign language generation, the model must convert the textual input into a sequence of signs. This requires not only generating the correct signs but also ensuring that the non-manual signals and the spatial grammar of the language are accurately represented.

6.2.4 Evaluation

Continuous evaluation with native sign language users should be conducted to gather feedback and improve the system. This involves usability testing, accuracy assessments, and iterative refinements based on user input. Moreover, metrics should be established for evaluating sign language models that account for the unique aspects of signed languages, such as the correct use of space and simultaneous gestures.

6.3 Other Ethical Concerns

Several groups stand to benefit from our model, while some may potentially be harmed. Students will primarily benefit from this model as it provides immediate and helpful responses to their academic inquiries, reducing their reliance on human assistants and allowing them to learn at their own pace. The model's 24/7 availability

offers continuous support, making it easier for students to get help outside regular hours, which is particularly beneficial for those with tight schedules or in different time zones. Educational institutions like EPFL will also see significant advantages. By reducing the number of student assistants needed, universities can lower operational costs and allocate resources to other areas.

However, there are potential harms associated with the model. One major concern is the accuracy of the responses. If the model generates incorrect or misleading information, it could hinder students' learning and lead to misunderstandings in their academic work. Overreliance on the AI tutor might also reduce students' critical thinking skills and their ability to solve problems independently.

Bias and fairness are other significant concerns. If the training data contains biases, the model could perpetuate these biases. Certain groups, especially those from underrepresented backgrounds, may be disproportionately affected by these biases, potentially receiving less accurate responses.

Privacy and security issues also need to be addressed. Ensuring that personal information is safeguarded is crucial. There is also the risk that the model could be exploited for purposes other than intended, such as generating inappropriate content or being used in ways that violate academic integrity.

To mitigate these risks, it is essential to improve the quality and diversity of the training data, ensuring it is representative of different backgrounds, perspectives, and problem-solving approaches. Clearly communicating the model's limitations to users and encouraging them to critically assess the AI-generated responses can help mitigate potential harms. Implementing mechanisms for users to provide feedback on the model's performance and flag incorrect or biased responses is also beneficial. Continuous monitoring and improvement of the model are necessary to ensure it remains accurate and fair, and robust privacy and security measures must be in place to safeguard student information, ensuring compliance with relevant data privacy regulations and best practices.

By considering these factors, we can maximize the benefits of our LLM while minimizing potential harms, ensuring that it serves as a valuable tool for enhancing education at EPFL.

7 Conclusion

Throughout this project, we developed a specialized Large Language Model tailored to EPFL course material, aiming to function as an AI tutor capable of providing detailed and timely responses to student inquiries without human intervention. Leveraging the T5-small model as our base, we enhanced its scientific question-answering capability through SFT, aligned its outputs with human preferences using DPO, and refined it further, using COT techniques, to generate concise, single-letter responses for multiple-choice questions.

Our main findings demonstrate significant progress in addressing the challenges faced by EPFL students. The final model achieved a notable MCQA accuracy of 32% on a dedicated testing dataset and 31.4% on the MMLU benchmark, surpassing random guessing despite its modest parameter count of 60 million.

Moreover, the successful implementation of advanced quantization techniques, such as LLM.int8(), facilitated a substantial reduction in model size from 242MB to 98MB while maintaining competitive accuracy at 29.99%. This highlights our ability to optimize model efficiency without compromising performance—a crucial advancement for deploying LLMs in resource-constrained environments.

However, our work also encountered limitations. The complexity and variability of EPFL course questions, the lack of available EPFL-tailored data, as well as the imbalance of classes in the MCQA training datasets, posed challenges in consistently delivering accurate responses, particularly in nuanced and complex topics. Furthermore, the model's performance in generating detailed explanations remains an area for improvement, possibly constrained by its relatively small parameter size. Scaling up the model's capacity (to higher number of parameters) and using a more balanced training dataset should be considered to address these limitations.

Looking forward, future work could explore several avenues. Applying the developed approach on larger models and enhancing the model's contextual understanding and reasoning capabilities through more extensive fine-tuning on diverse datasets could improve response accuracy across a broader range of topics. Additionally, refining the quantization process to handle outliers more effectively and exploring ensemble methods could further enhance model efficiency and robustness.

8 Contributions

- Jennifer Abou-Najm: data collection and pre-processing, supervised fine-tuning, direct preference optimization, report writing.
- Kelyan Hangard: data collection and pre-processing, supervised fine-tuning, direct preference optimization, hyper-parameters tuning and experimentation, quantization, specialization to MCQA format.
- César Camus-Emschwiller: data collection and

pre-processing, supervised fine-tuning, hyper-parameters tuning and experimentation.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Tim Dettmers, Mike Lewis, and Younes Belkada. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv preprint arXiv:2208.07339*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Jesse Dodge, Raghavendra Neeraja, Jason Kurtz, Matt Gardner, Nelson F. Liu, and Carissa Schoenick. 2021a. The english colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 837–850.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, and Matt Gardner. 2021b. Documenting the english colossal clean crawled corpus. Allen Institute for Artificial Intelligence.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

C.-Y. Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, page 74–81.

Pan Lu, Swaroop Mishra, Tony 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. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Muhammad Mateen. 2023a. Combining supervised and unsupervised learning for question answering in computer science theory. *Journal of Artificial Intelligence Research*, 67:245–269.

Mujtaba Mateen. 2023b. Computer science theory qa dataset.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, page 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 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. 2023a. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Rossen Rafailov, Zhengyuan Jiang, Sungmin Lee, and Graham Neubig. 2023b. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Flax Sentence Embeddings Team. 2021a. Stack exchange question pairs. https://huggingface.co/datasets/flax-sentence-embeddings/.

Flax Sentence Embeddings Team. 2021b. Stack exchange question pairs dataset. Retrieved from `https://huggingface.co/datasets/flax-sentence-embeddings/stackexchange-qp`.

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

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.