Flan-Tastic Learning: Enhancing University-Level Science Question Answering by Fine-tuning Flan-T5

Camille Nicole Gisèle Challier | 311020 | camille.challier@epfl.ch
Céline Kalbermatten | 328519 | celine.kalbermatten@epfl.ch
Wesley Elliott Stephen Monteith-Finas | 324745 | wesley.monteith-finas@epfl.ch
Chatbots-R-Us

Abstract

This report presents a comprehensive study on enhancing the performance of Flan-T5 models for university-level science questions. The primary aim was to develop an effective chatbot capable of answering first open-ended and later on multiple-choice questions (MCQs).

The contributions include the application of Supervised fine-tuning (SFT) to the pretrained model, the use of Direct Preference Optimization (DPO) to align the model with human preferences and quantization techniques aiming to enhance the computational efficiency of the model without compromising accuracy.

Key findings from the experiments demonstrate that fine-tuning the Flan-T5 model with task-specific prompts and optimal dataset orders results in substantial performance improvements in both zero-shot and few-shot setups. The integration of the DPO framework further refines the model's ability to generate human-preferred answers, while quantization methods provide a balanced approach to consume less memory and computing power.

This report highlights the potential of advanced fine-tuning and optimization strategies in developing robust Natural language processing (NLP) models for educational applications, paving the way for more efficient and effective AI-driven solutions in academic settings.

1 Introduction

The rise of Artificial Intelligence (AI) technology has made it easier to incorporate chatbot systems into education. This technology is increasingly being used for educational purposes (Okonkwo and Ade-Ibijola, 2021). However, developing intelligent chatbots capable of answering complex university-level science questions is a challenging task. Without specific fine-tuning or any other task-oriented approach, many current pretrained general-purpose language models struggle with it.

The current project aimed to address the mentioned challenge by applying several NLP methods such as SFT and DPO to pretrained models. The customised model was then further optimised to reduce its final size.

In this report, an overview of related work regarding the methods used to train the models and that guided decisions made throughout the project is provided. The chosen approach to develop an NLP model capable of handling complex university-level science questions is outlined, along with a detailed description of the data collection, evaluation methods, baselines and experimental details for each of the steps. Some ethical considerations are also thoughtfully analysed and included. Finally, the obtained results are presented and interpreted.

2 Related Work

SFT: To enhance the models' specialisation in answering university-level science questions and later in Multiple choice question answering (MCQA), instruction fine-tuning was employed. This process involves providing task-specific prompts to the model during fine-tuning, which typically enhances its performance. This practice is particularly beneficial for Flan-T5 models of all sizes, improving performance in both zero-shot and few-shot setups and enhancing scores on evaluation benchmarks such as the Measuring Massive Multitask Language Understanding (MMLU) (Chung et al., 2022) datasets. Consequently, recommended task-specific prompts for each fine-tuning step were used. Moreover, different orders in which datasets were presented during fine-tuning were experimented and their effectiveness compared. Indeed, some data orders are better than others and this can be observed even across tasks (Dodge et al., 2020). Consequently, the best performing order for each fine-tuning step was selected for further stages.

Evaluation Metrics: To assess the models' performance in open question answering, various evaluation metrics were employed. The Bilingual Evaluation Understudy (BLEU) score quantifies the similarity between a machine-generate answer and human-generated reference answers based on matching n-grams (Papineni

et al., 2002). The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score measures as well such overlaps using n-grams, word sequences or word pairs (Maskey and Hirschberg, 2005). However, these metrics often fail in nuanced evaluations requiring an understanding beyond lexical similarities. BLEURT score, which uses both supervised and unsupervised data to predict human judgment of text quality, outperforms the aforementioned traditional metrics. The main innovation of BLEURT is its ability to train with limited labelled data by using a large volume of unlabelled data. By learning from the patterns present in both types of data, the model can generalise better and becomes more robust against data scarcity and biases (Sellam et al., 2020).

DPO: The approach using the DPO Trainer of Hugging Face was heavily influenced by the work of (Rafailov et al., 2023), which introduces the DPO framework. It treats the constrained reward maximisation problem as a classification problem on human preference data, offering a stable, efficient, and computationally lightweight approach that eliminates the need for reward model fitting and extensive sampling.

Quantization: The quantization approach used in this project was inspired by the current advancements in quantization techniques for LLMs. Methods such as LLM.int8() (Dettmers et al., 2022), GPTQ (Frantar et al., 2023), and SpQR (Dettmers et al., 2023) have already been compared in previous studies (Jin et al., 2024). The strong performance and user-friendly implementation of LLM.int8() led to its adoption in the current project. LLM.int8() is a vector-wise quantization method, seamlessly integrated into bitsandbytes (BNB) within Hugging Face Transformers. This approach entails storing outlier submatrices in FP16 format while regular submatrices remain in int8. During matrix multiplication, separate computation is conducted for FP16 and int8 submatrices to safeguard outlier values. BNB also provides implementations of other powerful and state-of-the-art quantization techniques such as fp4, nf4, fp4-qd and nf4-qd, where qd stands for double quantization. The performances of these methods were compared across a spectrum of models, ranging from 3 billion to 70 billion parameters (Roy, 2023). The results indicate that in situations where the Graphics Processing Unit (GPU) memory is not limiting, prioritising bfloat16 usage is advisable for models up to 7 billion parameters. Alternatively, nf4 and fp4 are recommended to strike a balance between GPU utilisation, accuracy, and inference speed. Moreover, adopting double quantization methods may slightly decrease memory usage, although at the expense of inference speed. Additionally, further exploration with smaller models would be necessary to gain deeper insights. For the current project, methodologies based on established findings in the field were applied to such a smaller model.

3 Approach

To address the challenge of developing a model capable of answering university-level science questions, a dedicated pipeline was implemented. Initially, a pretrained model served as **base model**. This model underwent **domain-specific SFT**, followed by **DPO training** to refine its ability to generate human-aligned answers. Subsequently, another round of **SFT focusing on MCQA** format ensured accurate answer outputs. Finally, the model underwent **quantization** to reduce its size. Figure 1 illustrates each of these steps comprehensively.

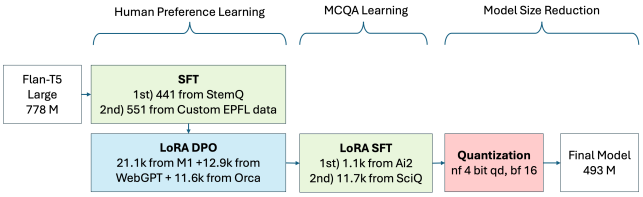

Figure 1: Schema illustrating the overall system pipeline. The different processes applied to the initial Flan-T5 large model can be seen leading up to the final SFT and DPO trained, and quantized model. For each step, detailed specifications of the datasets used and the number of training questions are provided. Additionally, information on the initial and final model sizes is included.

3.1 Data Collection

Preference Pairs Data: For the preference learning stage of DPO, a large amount of data was collected by presenting to ChatGPT 3.5 questions from École Polytechnique Fédérale de Lausanne (EPFL) courses combined with two different prompts. The two answers were then ranked using the following criteria: overall, correctness, relevance, clarity, completeness, conciseness and engagement. The entire Modern Natural Language Processing (MNLP) class of 2024 contributed to a portion of the dataset. In the following part, the personally selected prompting strategy is detailed.

Few-shot Learning: For the first prompt, a few-shot learning strategy was chosen (Dong et al., 2023; Qiao et al., 2023) heavily inspired by Liu et al's work (Liu et al., 2022). They suggest presenting and ordering each example based on the similarity between the token embeddings of the example questions and the token embedding of the real question. The embeddings were

calculated using the RoBERTa model, cosine similarity was used as the metric and the 3 most similar examples were presented in the prompt, ordered from least to most similar. As such, in an effort of clarity, only similar examples are presented, and potentially irrelevant ones are discarded. Regarding the examples, 22 of them were collected from EPFL courses available on Moodle, namely Statistical machine learning (MATH-412), Selected Topics in Cryptography, Computer security and privacy (COM-301), General physics I (PHYS-101(en)) and Machine Learning (CS-433).

Zero-shot Learning: For the second prompt, zero-shot learning was employed. Through experimentation of various instructions (Zhou et al., 2023), it was discovered that employing the simple directive "Let's think step by step" (Kojima et al., 2023) produced the most favorable answers.

In addition to the aforementioned dataset created by the MNLP class, other available datasets were also used. Further information regarding these datasets and their processing can be found in Section 4.1.

Open Questions & MCQA Datasets: For the SFT stages of the project, datasets with domain-specific questions and a golden answer, either an open-ended explanation or a single letter corresponding to one of the proposition, were used. This ensured that the models' responses align more closely with specific requirements and preferences, before and after DPO. The datasets used for this matter are explained in Sections 4.1.1 and 4.1.3.

3.2 Base Model: Flan-T5 Large

Flan-T5 is a variant of the T5 (Text-to-Text Transfer Transformer) model developed by Google, designed to perform various NLP tasks by converting all tasks into a text-to-text format (Longpre et al., 2023). T5 is known for its flexibility and efficacy across a range of tasks, from translation and summarisation to question answering and classification. Flan-T5 enhances the original T5 model with instruction fine-tuning, enabling it to better understand and follow specific instructions provided in the input. This makes Flan-T5 particularly effective for tasks that require understanding and executing complex instructions. It was considered a robust choice as a base model for fine-tuning in domain-specific applications, such as university-level science questions.

3.3 Human Preference Learning

Domain-specific SFT: Initially, the different Flan-T5 models underwent SFT to increase their performance in university-level science questions. Two distinct datasets within the domain were used. Each dataset was divided into training, evaluation and test subsets, with 80% allocated for training, and 10% each for evaluation and testing. This seemed to be a reasonable choice for the dataset and task in question (Muraina, 2022). Various combinations of arguments within Hugging Face's Seq2SeqTrainingArguments, also impacting the Seq2SeqTrainer, were tested. During training, a prompting strategy tailored for the Flan T5 model, appending "Question:" at the beginning and "Answer:" at the end of each question, was used. This prompt is proved to be effective for Flan T5 when general questions form the input (Chung et al., 2022). Initially, the Flan T5 small, large and base model underwent individual SFT on either the Custom EPFL course dataset or the StemQ dataset, both detailed in Section 4.1.1. Subsequently, the saved models were further fine-tuned with the remaining dataset. Since the order of datasets during fine-tuning is important (Dodge et al., 2020), both orders were compared. The models fine-tuned in the order resulting in a better performance were then chosen for further DPO training.

DPO Training:

The objective used for DPO training is defined as follows (Rafailov et al., 2023): $\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$

DPO can be viewed as fitting an implicit reward using an alternative parametrisation, where the optimal policy is simply π_θ. The gradient of the loss function $\mathcal{L}_{\text{DPO}}$ increases the likelihood of preferred completions y_w and decreases the likelihood of unpreferred completions y_l. The process is weighted by how incorrectly the implicit reward model orders the completions. This weighting is crucial for preventing model degeneration (Rafailov et al., 2023).

Low-Rank Adaptation (LoRA): After extensive testing, it was found that the DPO algorithm necessitated an extended training period and exhibited inconsistency in reducing the training loss. Consequently, LoRA was chosen as a solution. LoRA is an efficient and lightweight parameter reduction technique that significantly decreases the number of trainable parameters during fine-tuning (Hu et al., 2021).

The provided default sigmoid loss function was used for DPO training. Additionally, LoRA was incorporated with specific parameters for r, representing the rank of the update matrices, and $alpha$, representing the scaling factor. Furthermore, different configurations for the learning rate, batch size, and maximum sequence

length were tested to optimise performance, as detailed in Section 4.4.

Model Evaluation: In order to discriminate the best performing model at each stage, various task-specific evaluation metrics and datasets were employed. The specific methodologies are detailed in Section 4.2.

3.4 MCQA Learning

The DPO trained Flan-T5 large model underwent **SFT** to perform well in MCQA and to produce outputs in the desired format – specifically, only the letter corresponding to the correct answer. The used datasets consist of university-level science MCQs with predefined train, evaluation, and test splits. During training, a prompting strategy appending "Considering the different options, what's the best answer to this question:" at the beginning and "Letter of the answer:" at the end of each question, was used. This prompt effectively guides the model to output only a letter while prompting it to reason before making a choice. Derived from effective prompts for Flan T5 and the specific task at hand (Chung et al., 2022), this strategy ensured the desired optimisation. Various dataset orders for the Ai2 and the SciQ datasets, detailed in Section 4.1.3, were tested and compared. The best performing model, as determined by the **evaluation methods** outlined in Section 4.2, was subsequently fine-tuned further on the M1 MCQA dataset, detailed in Section 4.1.3. **LoRA** was used for all MCQA SFT steps to stabilise the training process (Hu et al., 2021).

3.5 Model Size Reduction

In the last step of the project, the goal was to reduce the final model's size for storage and memory advantages, while still retaining the same performance. Therefore, after the MCQA SFT phase (Figure 1), 4-bit and 8-bit quantization was applied using the BNB library (Dettmers, 2024). Experimentation with offloading, outlier threshold adjustments and nested quantization techniques were also done.

4 Experiments

4.1 Data

4.1.1 SFT

To begin with, **SFT** on domain-specific questions was applied using two datasets.

The **Custom EPFL courses** dataset was created by extracting questions from exercises and exams from 5 courses: Machine Learning (CS-433), General physics (PHYS-101(en)), Cryptography and Security (COM-401), Information security and privacy (COM-402) and Introduction to operating systems (CS-323). Python PDF parsing libraries, specifically PyPDF2 and PDFMiner, were used to automate the processing of course exams, thereby increasing the data pool. The dataset comprises a total of 551 entries, encompassing both open questions and MCQ with detailed answers.

The **StemQ** dataset (Drori et al., 2023) consists of 441 open questions from 18 of the 27 curated STEM university courses.

4.1.2 DPO

For the DPO training, three datasets related to human preference learning from relevant domains were used.

The **Milestone 1 (M1)** dataset has 1'522 unique exam questions along with different pairs of human preferred and rejected answers. Each question has on average 17.57 preference pairs and each of them was considered as a new data point. Samples where neither the chosen nor the rejected answer exceeded a length of 200 characters were removed, avoiding length biases and leading to 23'446 samples.

The **WebGPT Comparisons** (Nakano et al., 2022) dataset consists of 19'578 questions with two human scored answers each. It was processed by selecting the higher scored answers as preferred. Questions where both answers had the same score were removed, yielding 14'346 samples.

The **Intel's Orca DPO Pairs** dataset contains 12'859 examples from the OpenOrca dataset (Mukherjee et al., 2023), each with a chosen and rejected answer.

For each of the three former datasets, a 9:1 train test split was used.

The **Stanford Human Preferences** dataset (Ethayarajh et al., 2022) was only used for model performance evaluation. It contains 385'563 collective human preferences over responses to questions and instructions in 18 different subject areas from which only the four being related to university-level science questions were chosen. Each question has two answers with different scores. Based on these scores, the chosen and the rejected answer were determined. A total of 121'693 questions were built. The predefined test split of it contains the 5'980 questions used for evaluation.

To assess model performance, a test dataset, named **Performance** dataset, was created using the test splits of the DPO and StemQ datasets and treating the chosen answer from the pair of answers as golden answer. If a question appeared several times, the possible golden answers were stored in a list. This procedure reduced

the dataset from 8'656 to 3'337 unique questions.

4.1.3 MCQA

Four relevant datasets suitable for fine-tuning the model on MCQA were collected.

Allenai's SciQ dataset (Welbl et al., 2017) contains 13'679 science exam MCQs with predefined train, evaluation and test splits.

The **Ai2 Arc Challenge** dataset (Clark et al., 2018) contains 2'590 genuine grade-school-level, science MCQs, with predefined train, evaluation and test splits.

The **MMLU** dataset (Hendrycks et al., 2021) was processed to contain 2'401 MCQs from 13 out of the 57 STEM topics and was used only for testing purposes.

The **M1 MCQA** dataset was created from 792 unique exam questions in M1. The process of obtaining the ground truths from the preference pairs is detailed in Section A.3.

To assess model performance after MCQA fine-tuning and after quantization, a test dataset, named **MCQA Performance** dataset, was created using the test splits of the SciQ, Ai2, M1 MCQA and MMLU datasets.

4.2 Evaluation method

When evaluating the **SFT** and **DPO** trained models' performance, emphasis was placed on the F1, BLEURT and EM scores, although BLEU and ROUGE scores were also computed. The F1 score balances precision and recall and the EM score counts the number of questions where the predicted answer equals the label. As the BLEURT score cannot be interpreted on its own for a single model, it was calculated for each fine-tuned model and then compared with the one of the base model. The MCQA Performance dataset was used to calculate the percentages of questions where the trained model either had a higher or lower BLEURT score compared to the base model. By comparing all fine-tuned models to the base model, the best fine-tuned models could be identified. The F1 score for each model was calculated by averaging all individual question F1 scores. Additionally, test reward accuracies, corresponding to the mean of how often the chosen rewards are superior than the corresponding rejected rewards, were calculated at the end of each epoch on the DPO test datasets.

For the evaluation of the models fine-tuned on **MCQA**, the F1 score and the accuracy, measuring the proportion of correct predictions overall, served as evaluation metrics to compare the models' performances.

All evaluation metrics were consistently computed on the MCQA Performance dataset as well as its subsets. This approach was useful for analysis, as certain datasets were considered more significant than others.

4.3 Baselines

During the **SFT** stage, the fine-tuned Flan-T5 large model was compared to the non-fine-tuned model, as well as to other versions including Flan-T5 small, Flan-T5 base, and the BART model.

For the **DPO** stage, comparisons of the models with and without DPO, and also with the models before SFT were done.

In the **MCQA** stage, the model from the previous step and the original Google Flan-T5 large model were used as baselines.

The original and non-fine-tuned models served as the lower baseline throughout all comparisons. The goal was to ensure that the developed customised model consistently outperformed the original models at each stage of tuning.

4.4 Experimental details

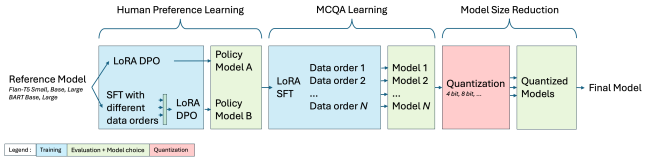

Figure 2: Schema illustrating the overall system pipeline. The diagram visualises the steps and model selections from the initial reference models to the final model.

Figure 2 provides a comprehensive overview of the system pipeline.

During **SFT training**, the emphasis was placed on experimenting with various dataset orders rather than hyperparameters, as the latter had less influence on the process compared to the dataset order. All SFT runs used a learning rate of 2e-5, a weight decay of 0.01 and 4 epochs.

Concerning the **DPO training**, different batch sizes, learning rates and maximum number of tokens were tested depending on the models. A limitation came from the tradeoff between the model size and the available GPU Random-access Memory (RAM). Therefore, large models needed a smaller batch size and maximum tokens. For the Flan-T5 small non-fine-tuned model, learning rates of 5e-5 and 1e-4 were tested, along with a maximum sequence length of 512 and a batch size of 8, balancing learning stability and computational efficiency. In the case of the Flan-T5 base non-fine-tuned model, combinations including a learning rate of 5e-5

and 1e-4 with a max length of 256 and batch size of 8, as well as with a max length of 512 and batch size of 4 were explored. For both previously SFT trained models, the best-performing hyperparameters from earlier experiments were employed, as there was a lack of time for further tuning. Due to the extensive running time of approximately 10 hours for the Flan-T5 large model, only a learning rate of 1e-4, identified as the best hyperparameter from the base and small models, was tested, with a max length of 512 and a batch size of 4. Additionally, the BART base model was trained using a learning rate of 1e-4, max length of 512, and batch size of 8, while the BART large model was configured with a learning rate of 1e-4, max length of 256, and batch size of 4. For all runs 3 epochs were used.

The **LoRA parameters**, with values of 16 and 32 for r and values of 32 and 64 for $alpha$, were tuned but showed very limited impact on performance.

During the **MCQA fine-tuning**, the hyperparameters employed in the SFT step and the LoRA parameters used in the DPO step were applied.

To optimise **quantization**, hyperparameters were meticulously tuned. For 4-bit quantization, parameters such as quantization data type were adjusted to enable 4-bit quantization. This involved substituting linear layers with FP4 or NF4 layers, alongside nested quantization techniques, aiming at significant memory reduction while preserving acceptable accuracy levels. Computational types such as torch.bfloat16 and torch.float32 were adjusted to optimise the model execution speed. In the realm of 8-bit quantization, outlier detection thresholds of 4, 6 and 8 were fine-tuned for robustness and accuracy. These adjustments aimed to strike a balance between precision and efficiency, enhancing the overall model performance.

4.5 Results

The test rewards accuracies of the different models after DPO training can be seen on Figure 3. The combination of the 10% test set splits of the M1, Orca and WebGPT datasets was used as the test set. Choosing BART as the base model led to less accurate reward predictions than the Flan-T5 models. Therefore, BART was removed from contention and not fine-tuned. Having an SFT stage prior to DPO increased the accuracy by $\sim 0.5\%$ for both the base and the large Flan-T5s. However, it decreased the accuracy of the small model by the same magnitude.

A learning rate of 1e-4 led to better results when tested on the Flan-T5 base model. Therefore, most of the subsequent runs used this value. The best performing models in all regimes have very similar test rewards accuracies and only deviate by $\sim 0.5\%$. Therefore, using solely this metric, no clear choice could be made on the model size to choose.

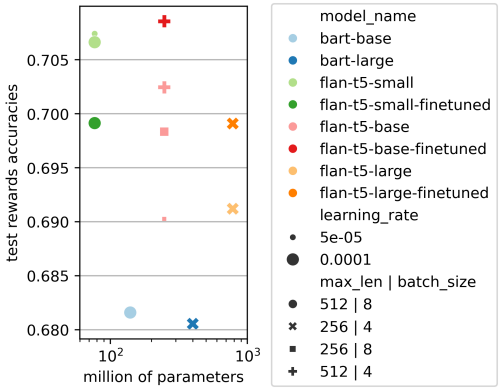

Figure 3: The test rewards accuracies obtained after the last training epoch for the different models and hyperparameters. The different models have very similar test rewards accuracies.

Building on these initial results, an additional analysis was done. The policy reward accuracies after DPO training on the Orca, M1 and WebGPT datasets as well as the F1 average scores and the EM scores on the MCQA Performance dataset of the different Flan-T5 fine-tuned and non-fine-tuned models were evaluated. Based on the found results, which can be visualised on Figure 11, the **fine-tuned Flan-T5 large** model was chosen as the best one after DPO training. It outperformed the other models in text generation quality, particularly in terms of the EM and F1 metrics.

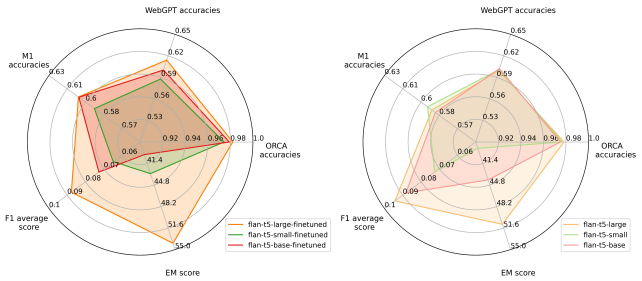

Figure 4: Spiderplot evaluating the Flan-T5 small, base, and large performances at each model's best training epoch. The models that underwent an SFT step prior to DPO are compared with the models that did not. The fine-tuned Flan-T5 large model performed best. The values for the BLEURT evaluations are available in the Appendix A.5.

The selected model, following DPO training, underwent various MCQA SFT combinations as outlined in 3.4. Subsequently, these models were evaluated on the MCQA Performance dataset and its subsets, focusing on F1 score and accuracy. As depicted in Figure 5, the initially fine-tuned model on the Ai2 dataset, followed by

a fine-tuning on the SciQ dataset, demonstrated a higher performance compared to other models fine-tuned differently. It achieved the highest overall accuracy and F1 score, enhancing both metrics by $\sim 1\%$ compared to the base model, Flan-T5 large. Consequently, the **Ai2-SciQ Flan-T5 large** model was selected as the final model before optimisation.

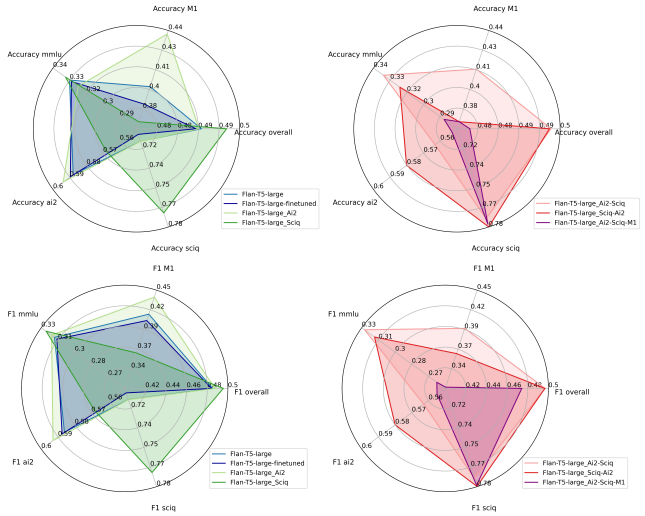

Figure 5: Spiderplot comparing the performances of MCQA fine-tuned models after DPO training. The Flan-T5-large model is the base model, the Flan-T5-large-finetuned the one after the SFT and DPO steps and the others the MCQA fine-tuned models. The results summarised on a table are available in the Appendix A.6. The Ai2-SciQ Flan-T5 large model performed best.

As a final step, various quantization techniques were applied to the Ai2-SciQ Flan-T5 large model. As depicted in Figure 6, 4-bit quantization using NF4 layers demonstrated better performance than both 8-bit quantization and 4-bit quantization using FP4 layers. Additionally, implementing double quantization further improved the results. The best performing quantization parameters were thus identified as **4-bit with NF4 layers, weights initialised with normal distribution, BF16 computation type, and double quantization**. As shown in Table 1, this quantized model significantly reduced the number of parameters and memory footprint and also improved the F1 score and accuracy compared to the initial model. The model's size and memory footprint were reduced by 36.6% and 68.88% respectively.

	overall F1	ai2 F1	sciq F1	mmlu F1	M1 F1	params	memory
Quantized Model	**0.499**	**0.570**	**0.777**	**0.329**	0.355	**493M**	**9.74e+08**
Initial Model	0.496	0.562	0.774	0.327	**0.395**	783M	3.13e+09
$\Delta_{quant-initi}$	0.00273	0.00767	0.00296	0.00196	-0.0405	-289M	-2.15e+09

Table 1: Performance comparison between the best performing quantized model, optained with 4 bits, nf4, bf16 computation type and double quantization, and the initial model after the SFT and DPO steps. The F1 scores, number of parameters and memory footprint are detailed.

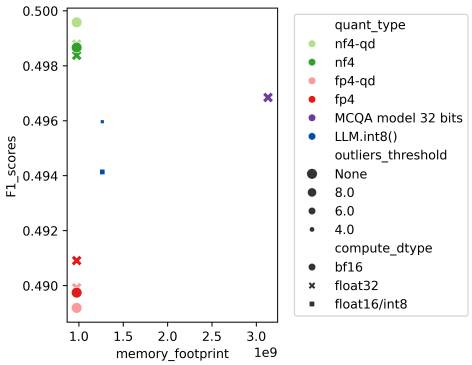

Figure 6: F1 scores obtained on the MCQA Performance dataset for various types of quantized models. The MCQA fine-tuned model served as reference model for the quantization. Different thresholds for outlier detection in llm.int8() and different computation types in matrix multiplication were used. The best performing quantization parameters were identified as 4-bit with NF4 layers, BF16 computation type, and double quantization.

5 Analysis

During the **DPO evaluation**, slight differences in average reward accuracies were observed between Figure 3 and Figure 11, likely due to inherent default parameters influencing the models' post-training behaviour. At this stage of the project, significant emphasis was placed on the M1 dataset, which was considered representative of the model's target task. Consequently, the performances on the M1 dataset were carefully analysed when selecting the best model after SFT and DPO.

However, the **MCQA fine-tuning phase** revealed that the M1 dataset may not be the most reliable. Further fine-tuning of the Ai2-SciQ model on the M1 MCQA dataset resulted in decreased performance across all subsets of the MCQA Performance dataset, including the M1 MCQA subset itself. This led to the conclusion that the M1 dataset should not be overemphasised in the model selection. Instead, models were predominantly evaluated based on their performance on the MMLU dataset, a well-established benchmark on which the models were not fine-tuned.

Analysis of performance across different subsets of the MCQA Performance dataset revealed notable differences in accuracies and F1 scores. As shown in Figure 5, the models generally perform well on the SciQ dataset, achieving average accuracies and F1 scores of approximately 75%. In contrast, the performance on the more challenging MMLU dataset is noticeably lower, with average accuracies and F1 scores around 32% and 31%, respectively. While the models perform better than random guessing, their effectiveness on highly complex university-level questions remains modest. Selecting a base model with more parameters could improve the

performance.

The Ai2-SciQ Flan-T5 large model, chosen as the final model before quantization, demonstrated improved performance over the base model, as previously mentioned. However, the performance gain was modest. Nevertheless, given the constraints of time and resources, these outcomes were within the expected limits.

In terms of academic performance, without considering the quality of the M1 dataset, it achieves a 41% accuracy rate, which falls below EPFL's passing grade of 60%. Therefore, it cannot be said that this model would be extremely beneficial for university-level educational purposes, as it lacks the competence to achieve high scores in exams.

As analysed in Table 1, the **quantization step** reduces the number of parameters and the memory footprint of the model, while still preserving, if not enhancing the overall performance. This leads to a smaller model which is easier to store, but also faster for inference. The increase in F1 score and accuracy is likely due to random variation, but it demonstrates that the model's performance is preserved even after quantization.

6 Ethical considerations

Adaptation to high- and low-resource languages: Flan-T5, the base model of the current work, is a multilingual model, which means that it already supports multiple languages. The proposed model could be further fine-tuned on domain-specific datasets in the desired languages other than English. High-resource languages as French or German benefit from abundant data, which can be used for both SFT and DPO, ensuring the model's robustness and accuracy. Low-resource languages as Urdu or Swahili have less available data and therefore several task cannot be trained directly in the specific language. However, the model could be adapted using cross-lingual transfer techniques, where a high-resource language model is used as a base, and the model is fine-tuned with smaller datasets from the low-resource languages (King, 2015). Data scarcity and quality issues are significant challenges for low-resource languages but techniques such as upsampling, data augmentation, and leveraging multilingual tokenizers can help mitigate these issues.

Adaptation to signed language: Incorporating multimodal learning approaches that include visual inputs can enable the model to interpret and generate signed languages. This would require training the model on datasets containing videos of signed language with corresponding translations (Yin et al., 2021). The development of comprehensive datasets featuring signed language paired with textual language would be necessary. Additionally, a specialised model architecture, possibly using convolutional or recurrent neural networks, would need to be designed to understand and generate signed language.

Possible benefits and harms: Users benefit from accurate, context-aware responses to university-level science MCQA questions. They can use the proposed model to study and solve exercises up to a certain level of question difficulty. Generally, the model is not meant for harmful uses. However, asking it harmful or discriminating questions could still lead to harmful answers. Fine-tuning the model with datasets annotated by toxicity experts could make it even safer and responsible (Raza et al., 2024).

Minority harms: Because the datasets used are diverse and do not focus on topics that could harm any minority groups, the potential harms apply equally to all users.

7 Conclusion

This project aimed to improve language models' performance in university-level science questions through experiments with SFT, DPO and quantization. The Flan-T5 large model, after DPO training and MCQA SFT on the Ai2 and SciQ datasets, outperformed other configurations. Including an SFT stage before DPO generally improved results, especially for larger models. Quantization techniques, particularly 4-bit quantization using NF4 layers, significantly reduced the model size and memory usage while maintaining performance. Evaluation highlighted the **Ai2-SciQ Flan-T5 large** model as the best one in terms of accuracy and F1 score on the MCQA Performance dataset. However, performance on complex datasets like MMLU showed room for improvement.

In conclusion, this project identified effective strategies using SFT, DPO, and quantization to enhance the performance of language models in university-level science question answering tasks. Future research could explore additional fine-tuning using diverse datasets, advanced quantization methods, and the extension of the same pipeline to larger models, aiming to enhance efficiency and accuracy.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, 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.

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.

Tim Dettmers. 2024. TimDettmers/bitsandbytes. Original-date: 2021-06-04T00:10:34Z.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. ArXiv:2208.07339 [cs].

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. ArXiv:2306.03078 [cs].

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey on In-context Learning. ArXiv:2301.00234 [cs].

Iddo Drori, Sarah Zhang, Zad Chin, Reece Shuttleworth, Albert Lu, Linda Chen, Bereket Birbo, Michele He, Pedro Lantigua, Sunny Tran, Gregory Hunter, Bo Feng, Newman Cheng, Roman Wang, Yann Hicke, Saisamrit Surbehera, Arvind Raghavan, Alexander Siemenn, Nikhil Singh, Jayson Lynch, Avi Shporer, Nakul Verma, Tonio Buonassisi, and Armando Solar-Lezama. 2023. A dataset for learning university stem courses at scale and generating questions at a human level. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: ACCURATE POST-TRAINING QUANTIZATION FOR GENERATIVE PRE-TRAINED TRANSFORMERS.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. 2024. A Comprehensive Evaluation of Quantization Strategies for Large Language Models. ArXiv:2402.16775 [cs].

Benjamin Philip King. 2015. *Practical Natural Language Processing for Low-Resource Languages.* Ph.D. thesis.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. ArXiv:2205.11916 [cs].

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Sameer Maskey and Julia Hirschberg. 2005. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. pages 621–624.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Ismail Muraina. 2022. Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback.

Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. 2021. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with Language Model Prompting: A Survey. ArXiv:2212.09597 [cs].

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:2305.18290 [cs].

Shaina Raza, Oluwanifemi Bamgbose, Shardul Ghuge, and Deepak John Reji. 2024. Safe and responsible large language model development.

Somnath Roy. 2023. *Understanding the Impact of Post-Training Quantization on Large Language Models*.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions.

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

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large Language Models Are Human-Level Prompt Engineers. ArXiv:2211.01910 [cs].

A Appendix

A.1 AI usage

For this project, various AI tools were used to support the work.

- **Copilot** was activated in VS Code to speed up the creation of code.
- Occasionally **ChatGPT** was also used to get ideas for parts of the code, especially for certain plots or data processings.

For most parts of the project such as SFT, DPO or LoRA, the examples from Hugging Face usually proved to be more useful than the suggestions from ChatGPT. Therefore, only Copilot was mainly used for the code.

Copilot's suggestions were easy to review as they focused mainly on syntax and less on content. The code suggestions from ChatGPT were only used as a starting point and were edited further. This ensured an appropriate and controlled use of the AI tools.

A.2 Group collaboration

- **Design of model and training strategies:** Camille Challier, Céline Kalbermatten, Wesley Monteith-Finas
- **Dataset development:** Camille Challier, Céline Kalbermatten, Wesley Monteith-Finas & MNLP Class of 2024
- **Implementation of training infrastructure:**
  - **SFT**: Céline Kalbermatten
  - **DPO**: Camille Challier, Wesley Monteith-Finas
  - **MCQA SFT**: Céline Kalbermatten
  - **Quantization**: Camille Challier, Wesley Monteith-Finas
  - **Quantitative model evaluation**: Céline Kalbermatten
  - **Visual model evaluation (plots)**: Camille Challier, Wesley Monteith-Finas
- **Report:** Camille Challier, Céline Kalbermatten, Wesley Monteith-Finas
- **Coordination of results and model analyses:** Camille Challier, Céline Kalbermatten, Wesley Monteith-Finas
- **Research Advisors:** Mentor Ahmet Sencan, Professor Antoine Bosselut

A.3 Extracting MCQAs from M1 Data

For each preference pair, if one response was marked correct, a cosine similarity was calculated between the sentence embeddings of the answer and each proposition. The highest scoring proposition received a vote. After all preference pairs were processed, the proposition with the most votes became the ground truth answer. This method is only accurate if the "correctness" annotation was fact-checked and has a limitation when multiple propositions are correct, as it only selects the most voted one. The embeddings were calculated using the *TfidfVectorizer* from the Sci-Kit Learn python library.

A.4 DPO Training

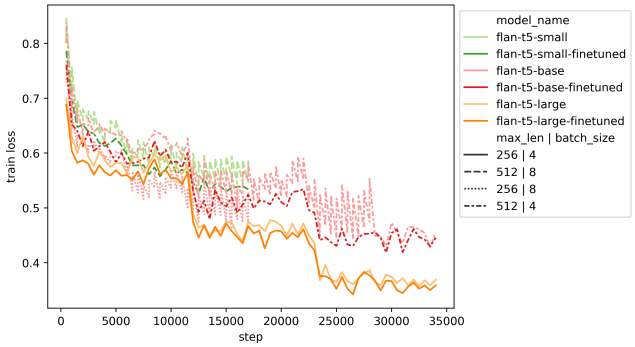

Figure 7: Train loss by training step. BART losses are omitted because they are not of the same magnitude.

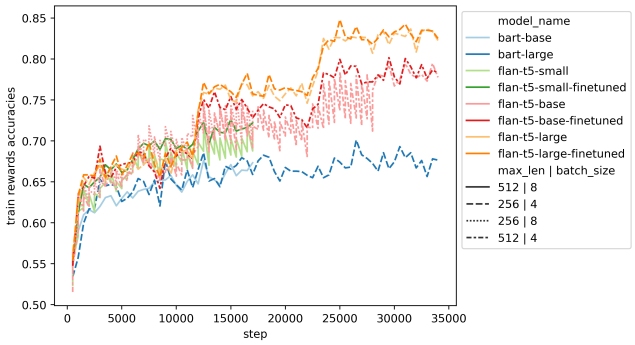

Figure 8: Train rewards accuracies by training step.

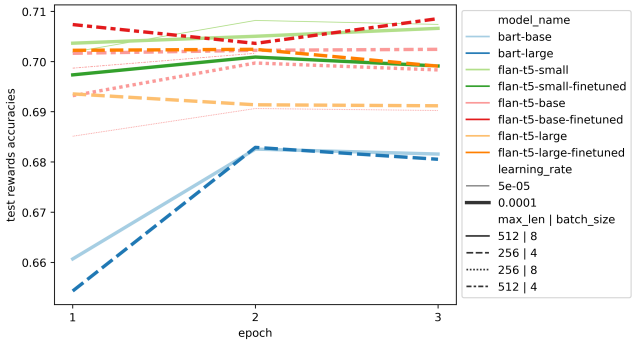

Figure 9: Test rewards accuracies by training epoch.

A.5 Model performance evaluation

The table on *Figure 10* compares the BLEURT score of the different trained Flan T5 models with the base models. The base models are the original models without any DPO or SFT training. For each of the 3'337 questions of the performance evaluation dataset, a BLEURT score was computed for both the trained and the base model. In the table, the percentages out of all 3'337 questions when the trained model attained a higher or a lower BLEURT score than the base model are shown. In case of equality, the question was not taken into account.

	Bleurt > base model	Bleurt < base model
Flan T5 small DPO	0.373089601	0.253221456
Flan T5 small SFT DPO	0.274198382	0.295175307
Flan T5 base DPO	0.443512137	0.250224753
Flan T5 base SFT DPO	0.352412346	0.320647288
Flan T5 large DPO	0.450704225	0.227449805
Flan T5 large SFT DPO	0.397362901	0.263110578

Figure 10: Table showing the percentages of questions where the trained model achieved a higher or lower BLEURT score than the corresponding base model. The data illustrates that across nearly all trained models, the BLEURT score surpasses that of the base model more frequently than it falls short. This trend strongly suggests the utility of the SFT and DPO trainings.

A.6 MCQA SFT evaluation

	F1 overall	F1 M1	F1 mmlu	F1 ai2	F1 sciq	Accuracy overall	Accuracy M1	Accuracy mmlu	Accuracy ai2	Accuracy sciq
Base model	0.485229773	0.415805379	0.317328702	0.586093353	0.717992722	0.488878698	0.4	0.328117591	0.587883959	0.719
M2 model	0.484150592	0.406700627	0.315044976	0.587500234	0.71318281	0.487273561	0.3875	0.326695116	0.589590444	0.714
Ai2	0.487440615	0.440091036	0.320103761	0.592954187	0.718069364	0.488190782	0.4375	0.321953532	0.593856655	0.719
Sciq	0.495531607	0.360621315	0.32535523	0.568104044	0.769928423	0.498216464	0.375	0.330962541	0.568259386	0.77
Ai2-sciq	0.496843405	0.395600928	0.327305727	0.562934896	0.774062342	0.49736299	0.4125	0.332859175	0.563139932	0.774
Sciq-ai2	0.496608615	0.35966954	0.317728713	0.580029114	0.779812186	0.49690438	0.375	0.321005216	0.580204778	0.78
Ai2-sciq-M1	0.47420243	0.311814189	0.258049872	0.554880805	0.778984971	0.473744554	0.375	0.289236605	0.552901024	0.778

Figure 11: Table comparing the F1 scores and accuracies for the different MCQA SFT combinations. The initially fine-tuned model on the Ai2 dataset, followed by a fine-tuning on the SciQ dataset, demonstrated a higher performance compared to other models fine-tuned differently.