Progress Report: AI-Tutor Specialized to Course Content at EPFL

Stefan Krsteski | 370315 | stefan.krsteski@epfl.ch
Matea Tashkovska | 370319 | matea.tashkovska@epfl.ch
Mikhail Terekhov | 370164 | mikhail.terekhov@epfl.ch
Said Gürbüz | 369141 | said.gurbuz@epfl.ch
chatterbox

1 Introduction

The primary goal of our work is to enhance Phi-3-mini-4k-instruct (Abdin et al., 2024) using Direct Preference Optimization (DPO) (Rafailov et al., 2024), focusing on fine-tuning to align with human preferences. In this progress report, we discuss 5 main sections: Datasets, Model, Preliminary Training Results and two Model Adaptations.

2 Dataset

DPO Datasets

To develop our DPO model, we explored two distinct sources for training data: the provided dataset from Milestone 1 and preference data found online.

The provided dataset collected for Milestone 1 included 1522 unique questions, each featuring multiple preference pairs ranked on various ranking criteria. We determined the chosen and rejected answer for each preference pair of every question based on the 'overall' rating. This resulted in more than 20,000 entries, each comprising a question with a rejected and chosen answer. We decided to focus exclusively on the 'overall' ranking since we found it to be a more consistent and reliable measure of preference compared to individual criteria, which can vary significantly between raters due to their subjective nature. The overall ranking reflects the user's combined opinion on all other criteria, which aligns with our end goal of predicting outcomes that best meet user preferences.

To further enhance our model, we used data found online. More specifically, we use two widely used preference datasets - Stack Exchange (Team, 2021) and Ultra Feedback (Cui et al., 2023).

The Stack Exchange dataset includes questions along with their most upvoted and most downvoted answers from the Stack Exchange network, which we use as preference pairs of chosen and rejected answers, respectively. The dataset covers a variety of domains, but we filtered it to contain only subjects that are relevant for this project. More specifically, we focused exclusively on STEM categories, making a total of 54,458 entries. The detailed list of these categories is included in the appendix.

Our second dataset found online - Ultra Feedback, comprises prompts from diverse resources (like UltraChat, ShareGPT etc.), each one with 4 different responses generated by different LLMs. Each response is rated using GPT-4 based on fine-grained criteria including instruction-following, truthfulness, honesty, and helpfulness. We decided to use the already binarized Ultra Feedback dataset [1] which was obtained by preprocessing the original dataset. During this preprocessing, the response with the highest mean rating out of the four was selected as the chosen answer, and a response with a lower mean was randomly chosen as the rejected answer. The final UltraFeedback dataset consists of 60,917 entries.

For all of the above-mentioned datasets we removed entries where both the chosen and the rejected answer were the same to avoid confusing the model. Finally, all datasets were formatted to follow the format specified in the project description.

MCQA Datasets

We preprocessed two datasets for the MCQA task. The Measuring Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) dataset contains a diverse collection of multiple-choice questions from various fields. This dataset includes an auxiliary train split with 99,842 entries for training and a test split with 14,042 questions from different subjects for testing our model.

The second dataset that we will use is the AI2 Reasoning Challenge Dataset (ARC) (Clark et al., 2018), which comprises 3,370 multiple-choice science exam questions for training, sourced from various origins, along with 869 questions for validation and 3,548 for testing.

[1] https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences

These datasets are well-suited for our goal of developing a real-world LM for educational assistance, as they consist of STEM-related questions with precisely four answer choices, one of which is correct. All of the datasets have been formatted in accordance with the specifications outlined in the project description.

3 Model

We used Phi-3-mini (Abdin et al., 2024) model as the base architecture - a decoder only model with 3.8 billion parameters.

The pre-training of Phi-3-mini was conducted on a massive corpus consisting of 3.3 trillion tokens sourced from diverse, high-quality datasets. This corpus includes heavily filtered web data and synthetic LLM-generated data, ensuring the model learns from both human and machine-generated content. The focus was on educational and reasoning skills, aligning perfectly with our goal of creating an AI tutor.

The post-training process of the Phi-3-Mini model includes Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). SFT utilizes high-quality, domain-specific data such as math, coding, reasoning, and conversation to enhance the model's capabilities. Following SFT, DPO aligns the model with human preferences by steering it away from unwanted behaviors using labeled data. This phase focuses on chat formats, reasoning tasks, and responsible AI.

Loss Function

For training, we used the sigmoid loss on the normalized likelihood with the logsigmoid, which is proposed in the original DPO paper (Rafailov et al., 2024). This method aligns the model with human preferences by optimizing the log probability of preferred responses over dispreferred ones. The loss function is defined as follows:

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

In this formulation, π_θ is the policy model, π_{ref} is the reference model, y_w is the preferred response, y_l is the rejected response, β is a scaling hyperparameter, and σ is the sigmoid function. This objective encourages the model to increase the likelihood of preferred responses while decreasing the likelihood of dispreferred ones, ultimately aligning the model outputs with human preferences.

4 Preliminary Training Results

Our training approach leverages LoRA adapters (Hu et al., 2021), fine-tuning our Phi-3-mini model efficiently. We used a β value of 0.1, in line with the official DPO paper. We used AdamW optimizer with a learning rate of $5\text{e}{-5}$ and a linear scheduler to stabilize training. The model was trained with batch size of 1 across three epochs and enabled $FP16$ training to leverage faster computations and reduce memory usage. The LoRA configuration includes alpha set at 16, the rank of the update matrices (r) set at 32, and a dropout of 0.1. The list of target modules to which LoRA layers were applied included q_proj, k_proj, v_proj, o_proj, $down\_proj$ up_proj, and $gate\_proj$. The max length was set to 512 for the chosen and the rejected answer, and 128 for the prompt. For our experiments, we tested four different data combinations to identify the optimal mix for training our model. We set aside 10% from all DPO datasets for constant evaluation across all experiments. From Table 1 we can see the results for the different data combinations. As expected, training on all the datasets yielded the highest validation reward/accuracy of 83.36% and lowest validation loss of 0.476. Therefore, since our goal is a model aligned to the preferences captured with our data, we choose the model trained on all three datasets as our final model. The training loss curve for our final model be found in the appendix, Figure 1.

Furthermore, to evaluate the overall generative ability of our final model, we utilized MT-Bench (Zheng et al., 2024). MT-bench evaluates LLMs using GPT-4 as a judge, through multi-turn conversations, focusing on their ability to engage in coherent and engaging exchanges. We used two of the benchmark's strategies: single-answer grading and pairwise win rate. Using the single mode, we positioned our Phi-3 Mini DPO on a competitive leaderboard among various advanced models, achieving an impressive average score 8.2 out of 10, with higher score on STEM than GPT-4. In the pairwise mode, we directly compared our chosen policy model against the base model with GPT-4 serving as the judge, to try and gauge the impact of DPO on aligning model responses. The results show that our Phi-3 Mini DPO recorded 23 wins, 20 losses, and 37 ties against the base model. These results suggest that the final model is effective in understanding complex queries, making it a competitive option among state-of-the-art language models.

EPFL Preference	UltraFeedback	Stack Exchange	Eval Acc. (%) ↑	Eval Loss ↓
✓			55.00	0.696
✓	✓		75.47	0.624
✓		✓	79.06	0.516
✓	✓	✓	**83.36**	**0.476**

Table 1: Performance of DPO Trained Phi-3 Mini Model with Different Training Dataset Combinations

This evaluation may suffer from "positional bias", an effect we discuss in Appendix A.3.2. There we also show that the model after DPO still performs better when this is accounted for. Further insights into category-specific performance and comparative analysis with other models are presented in the appendix.

5 Retrieval-augmented generation

For retrieval-augmented generation (RAG) we will use the wikipedia dataset from Huggingface [2]. To avoid going through the whole dataset, we will manually select entries, which are the most relevant to a subset of questions in ARC. We will pre-compute embeddings of passages from this selection generated by Phi-3 (the average features from the last hidden layer of the model). At inference time, we will compute the embedding of the query and fill the context with passages with the closest embeddings. This procedure does not require us to change the loss nor to update the architecture. In addition to the strategy from this milestone, we will consider evaluating our RAG model on both of our MCQ benchmarks to compare it to our non-augmented model.

6 Quantization

As detailed in our proposal, we will explore both Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) without the need for additional datasets, as we believe we have collected a sufficient amount by now. Given our ongoing work with LoRA, we are confident in our ability to integrate INT8 quantization, which will offer an optimal balance of performance retention and model size reduction. Current evidence suggests that there is no immediate need to modify the loss function specifically for quantization. For its evaluation we will use the exact same strategy used in this milestone. Additionally we will consider comparing the quantized and non-quantized model on both of our MCQ benchmarks.

[2] https://huggingface.co/datasets/wikipedia

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

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

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. *arXiv preprint arXiv:2106.09685*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Asa Cooper Stickland and Iain Murray. 2020. Diverse ensembles improve calibration. *arXiv preprint arXiv:2007.04206*.

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

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Categories Used from the Stack Exchange Dataset

We filtered the Stack Exchange dataset to include only entries from the following subjects: physics, bioinformatics, electronics, mathoverflow, codereview, cs, cstheory, datascience, matheducators, engineering, ai, cseducators, iot, softwareengineering, stats, networkengineering,scicomp, robotics, devops, astronomy, askubuntu, apple, serverfault, security, webapps and webmasters.

A.2 The Training Loss Curve for our Final DPO Model

Figure 1: Training loss curve for the best model

Note that we had to use batch size of 1 to accommodate the model in the memory, which resulted in a highly stochastic loss function. Three epochs of training are clearly visible in the curve. Apart from the drops between epochs, a clear fast learning trend is visible in the beginning of the first epoch.

A.3 MT-Bench Evaluation

As outlined in the main text, we evaluated our final DPO model on the MT-Bench benchmark, using single-answer grading and pairwise win rate modes. Single-answer grading based on GPT-4 can rank models effectively and matches with human preferences well.

A.3.1 Single Answer Grading

In Table 2, we present the MT-Bench as a column on the leaderboard based on single-answer grading with GPT-4. This result is impressive, considering the size of the model, 3.8B parameters. This suggests that our model is highly capable of understanding queries and responding in a way that humans prefer. To be fully objective, we also included the reported result from the Phi-3 on MT-bench, which can be found in their technical report

Model	Score
GPT-4	8.99
Phi-3 Mini	8.38
Phi-3 Mini DPO	**8.20**
GPT-3.5 Turbo	7.94
Claude v1	7.90
Vicuna 33B v1.3	7.12
Llama 2 70B Chat	6.85

Table 2: The breakdown of LLMs' MT-bench scores in the average from 1st and 2nd turn of a dialogue. Full score is 10.

(Abdin et al., 2024). Unfortunately, the authors do not include further relevant details, such as the fine-grained performance. Considering this, it is difficult to directly compare our model to theirs using this benchmark mode without the per-subject scores, since we focused on fine-tuning for STEM-related topics.

To further assess our final DPO model's capabilities, we were interested in seeing its fine-grained performance, even more so than the aggregated overall score. This interest stems from the fact that our model is aligned on datasets mainly consisting of STEM questions. The comparison of 4 representative LLMs regarding their abilities in 8 categories: Writing, Roleplay, Reasoning, Math, Coding, Extraction, STEM, Humanities, shown in Figure 2.

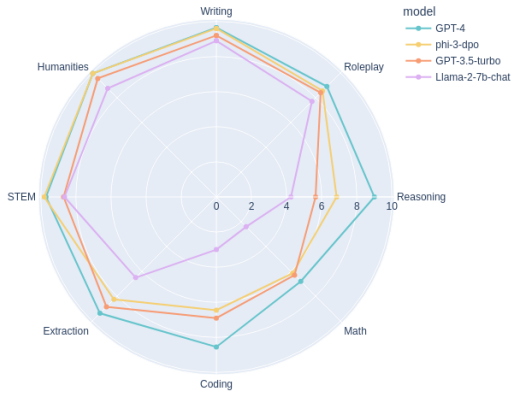

Figure 2: Comparison of our DPO model against 3 state of the art models on the MT-Bench across 8 categories

As we can see, the STEM score of 9.8, is higher than GPT-4's, which shows that our model is highly capable for queries from this domain. Another

valuable insight from this graph are the weaknesses, such as Coding and Math, which we should aim to improve for Milestone 3.

A.3.2 Pairwise Win Rate - Positional Bias

Match-up	A Wins	B Wins	Ties
Phi-3-DPO vs. Phi-3	126	62	8
Phi-3 vs. Phi-3-DPO	115	76	8

Table 3: Comparison Results for Phi-3-DPO and Phi-3 (base) using GPT-3.5 as a judge

Yet again we attempted to do a head-to-head comparison of our final DPO model with the base model. In order to do this we decided to try using the pairwise comparison. The comparison is done using GPT-4 as a judge to asses which is the better response, this method is explained in details in the Judging LLM-as-a-judge paper (Zheng et al., 2024).

Even though we obtained the results mentioned in the main text with pairwise-comparison using GPT-4, we decided to investigate this mode further. Since MT-Bench is an expensive toy, and we are a group of students, we decided to reproduce the judging from the official repository, this time using the provided GPT-3.5 API from Milestone 1, instead of GPT-4 (as in the original benchmark). We want to highlight an important pitfall of using this method - the positional bias. To show this we conducted two experiments. First we sampled 200 questions from the given Milestone 2 example dataset. In the initial experiment the first given responses were from the DPO model, while in the second experiment we swapped the positions. With Table 3, we demonstrate how the position of queries can significantly influence judgments, where the first given answer was preferred most of the time. Even though there are possible ways of mitigating the positional bias, outlined in (Stickland and Murray, 2020) and (Wang et al., 2023), the original method does not utilize them, marking this mode as unreliable.

Importantly, an essential insight emerges from our experimentation: an awareness of this bias and its potential impact on the MCQA evaluation for the final milestone. Recognizing that our model likely suffers from this bias, we are committed to implementing strategies to mitigate its effects and enhance the overall performance of the final MCQA system.