ProbLLMs: Multiple-Choice Problem Solving for EPFL Courses

Arvind S. Menon | 354584 | arvind.menon@epfl.ch
Lars C.P.M. Quaedvlieg | 352130 | lars.quaedvlieg@epfl.ch
Lazar Milikic | 353622 | lazar.milikic@epfl.ch
Somesh Mehra | 353628 | somesh.mehra@epfl.ch
GradientVanishers

1 Introduction

We aim to develop an AI tutor focused on **multiple-choice question-answering** for EPFL courses. We will explore both LLaMA-based and DeepSeek-based models, assessing their efficacy through complex reasoning benchmarks. Our approach includes DPO-driven fine-tuning for smaller models and LoRA-based tuning for larger ones. We plan to utilize RAG with Vector and Graph-based query engines and exploit SmoothQuant quantization for efficiency improvements.

2 Data

The initial dataset comprises approximately 30K aggregated data points annotated by MNLP students, which serve as the primary training data for DPO. We will preprocess this dataset to extract only high-quality data points. These are defined as entries with different explanations where at least one is correct. Specifically, we will filter out data points that have an empty correctness field or where four or more relevant fields are filled with the answer "AB.", ensuring that we filter out ambiguous, incorrect or very similar pairs.

Our focus is to specialize the model for Computer Science, Machine Learning, and Physics courses. Thus, we intend to augment our dataset with problem-solution pairs relevant to these disciplines. For Computer Science, we will utilize a publicly available Kaggle dataset (Mateen, 2023), including question-answer pairs from 150 subtopics.

Additionally, the TheoremQA dataset (Chen et al., 2023) offers 800 QA pairs annotated by experts, covering over 350 theorems across Mathematics, Computer Science, and Physics. This dataset's high-quality annotations could be particularly beneficial for training our model to comprehend and apply theorems effectively when answering numerical questions.

For theoretical physics, we plan to incorporate the CAMEL AI Physics Questions dataset (Li et al., 2023), which contains 20K problem-solution pairs from 25 physics topics generated using GPT-4. We will also utilize the CAMEL AI Math dataset, which includes math questions accompanied by correct answers and explanations to improve mathematical reasoning.

As our approach requires preference pairs for data to apply to DPO, we propose two methods to automatically annotate a large set of preference pairs across the selected datasets. Initially, we will employ our most effective prompting strategy from milestone 1 to generate two independent answers of comparable quality for each question within the datasets. Our confidence in this strategy stems from observations made during annotation, where it consistently yielded two comparable high-quality responses. To determine which response is superior, we will re-apply ChatGPT (Gilardi et al., 2023) by prompting it to assess the correctness and completeness of the answers relative to the dataset's reference solutions. We employ a few-shot learning approach, leveraging examples that we have previously manually annotated, to further ensure the quality of the labels. Additionally, we plan to explore a RoBERTa-based approach as described in appendix A.

3 Methodology

3.1 Generator Model

We plan to fine-tune and evaluate three models:

DeepSeekMath-7B (Shao et al., 2024) A mathematics-specialized model, pre-trained using a variety of mathematical resources.

Llama3-(8B, 70B) (Touvron et al., 2023) A series of language models trained across diverse domains recognized as the best-performing open-source options.

Although Llama3-8B consistently outperforms other open-source models, DeepSeekMath-7B ex-

cels in mathematics, rivaling the performance of Llama3-70B. Moreover, recent studies suggest that LLMs derived from models pre-trained on mathematics or coding tasks demonstrate enhanced reasoning capabilities (Shao et al., 2024; Azerbayev et al., 2023). Performance comparisons of these models are detailed in Appendix Table 1.

3.2 Fine-Tuning Strategy

We have access to at least one NVIDIA A100 80GB (NVI, 2024) and will employ Direct Preference Optimization (DPO) (Rafailov et al., 2024) for fine-tuning our language models. We plan to attempt full-parameter fine-tuning for Llama3-8B and DeepSeekMath-7B, depending on available compute, but primarily aim to utilize Low-Rank Adaptation (LoRA) (Hu et al., 2021) for parameter-efficient tuning. For the larger Llama-70B model, we may attempt quantized LoRA (Dettmers et al., 2024). These methods will be implemented using HuggingFace's PeFT (Mangrulkar et al., 2022) and TRL (von Werra et al., 2020),

3.3 Quantization

To reduce the memory requirements and inference time for our model, we will apply post-training quantization using the SmoothQuant method (Xiao et al., 2023). This method enables 8-bit weight 8-bit activation (W8A8) quantization, and they showed minimal loss of performance with significant speedups and lower memory footprints for a variety of LLMs, thus it seems like a suitable choice for our purposes.

3.4 Retrieval Augmented Generation

To reduce hallucinations and ground the generations using factual information, we will implement Retrieval Augmented Generation (RAG). We will leverage resources for relevant courses from the EPFL IC Drive, Moodle, and additional textbooks available online (Open Education Network, Accessed 2024) to build our knowledge base. Specifically, we will experiment with building both a traditional vector database and a knowledge graph index (KGI). The former is a reliable and proven approach for RAG, whilst the latter, though less widespread, could be more effective for our domain given the highly structured nature of course materials; a knowledge graph could better capture relationships between concepts. (Gao et al., 2023).

For implementation, we plan to primarily use LlamaIndex (Liu, 2022) for building, querying and storing our vector database and KGI.

We will evaluate the performance using the same method as for the un-augmented model, as specified in section 4.

4 Evaluation

We plan to assess our model's performance using established multiple-choice reasoning benchmarks suited for complex EPFL course content. The datasets include: **GSM8K:** A collection of 8,500-grade school math problems; **MATH:** A set of 12,500 competition-level math problems; **MMLU:** Diverse tests across 57 STEM, humanities, and social sciences topics; **BBH:** Challenges comprising 23 advanced tasks in multi-step reasoning. Additionally, we will utilize a hold-out set of the cleaned preference data obtained from students.

For quantitative evaluations, these benchmarks are well-suited for language models, including our planned enhancements with quantization and RAG. Particularly, the EPFL dataset will serve as the primary benchmark, employing the pre-trained base models from section 3.1 as baselines. We will instruct the model to output their final choice inside a \boxed{} to facilitate evaluation.

To evaluate open-ended questions from MNLP students-annotated preference data that is held out, we plan to apply BERTScore (Zhang et al., 2019) to compare the answers produced by our models with extracted 'correct' answers from the dataset.

Qualitatively, we will employ few-shot in-context prompting to gather multiple-choice answers from the models. Each evaluator will compare outputs from the baseline and enhanced models against a randomly selected set of questions, enabling us to gather diverse assessments of model performance.

5 Ethical Considerations

AI systems capable of solving exam questions could facilitate cheating if students use them as shortcuts to bypass studying, abusing educational integrity and depriving students of learning experiences. Additionally, these models often demand huge amounts of computational resources, widening the disparity between well-funded and under-resourced schools.

To mitigate the risk of AI-facilitated cheating and resource disparities, institutions can implement API access controls and provide equitable technology access programs.

References

2024. NVIDIA A100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/a100/. Accessed: 2024-05-08.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. Theoremqa: A theorem-driven question answering dataset.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

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*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society.

Jerry Liu. 2022. LlamaIndex.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Mujtaba Mateen. 2023. Computer science theory qa dataset.

Open Education Network. Accessed 2024. Open textbook library. University of Minnesota.

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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Language Model	Knowledge	Arithmetic	Math	Reasoning
	MMLU	GSM8K	MATH	BBH
Llama3-8B	68.4	79.6	30.0	-
Llama3-70B	**82.0**	**93.0**	50.4	**80.1**
DeepSeekMath-7B	-	88.2	**51.7**	-
Mixtral 8x22B	77.8	87.9	49.8	78.4
Qwen1.5 72B	76.2	81.9	40.6	65.9

Table 1: Comparison of open-source Language Models across complex reasoning benchmarks

A Automated preference labelling

We also plan to explore an alternative approach involving the use of a masked-language model to determine which of the two ChatGPT-generated solutions is more appropriate. Specifically, we propose fine-tuning RoBERTa (Liu et al., 2019) using student-annotated preference data. In this methodology, the input to the model will consist of the question concatenated with two potential answers, labeled as A and B.