MNLP Project Proposal

André Santo | 376762 | andre.loureiroespiritosanto@epfl.ch
Pedro Chaparro | 374339 | pedro.chaparro@epfl.ch
Pierre Høgenhaug | 385070 | pierre.hogenhaug@epfl.ch@epfl.ch
My **N**ice **L**ong **P**enguin

1 Introduction

In our project, we aim to create an academic chatbot using open-sourced large language models like Phi-2 and Flan T5-XL. Our goal is to make a chatbot that can correctly answer technical academic questions. We will improve our chatbot by using Supervised Fine Tuning (SFT), followed by Direct Preference Optimization (DPO), which helps the chatbot learn from examples and feedback. This approach is inspired by successful models like InstructGPT and aims to make our chatbot perform better by learning from real interactions.

2 Model

We will begin with a pre-trained model and apply SFT to adapt it to our specific task of answering multiple-choice questions (MCQs). Subsequently, we will implement a regularized version of DPO that controls for length bias exploitation, achieving better performance when controlled for length, as done in the work of (Park et al., 2024). To implement SFT, we consider 2 options:

Update model weights We can update all of the model weights using SFT with next token prediction. This is the most straightforward way to perform SFT. However, this might not be desirable due to the computational cost and because some of the model knowledge acquired from the massive pre-training might vanish.

Use an adapter Adapters allow us to tweak small parts of the model specifically for our tasks without altering the entire model structure. By using an adapter, we have to initialize new Feed-Forward Network (FFN) layers between components of transformer blocks. Only these layers are then updated. This decreases the computational cost and preserves the pre-trained model weights. One type of adapter that can be used is the LoRa adapter (Hu et al., 2021) or COMPACTER adapter (mahabadi et al., 2021). A framework to do so is presented in (Dettmers et al., 2023).

2.1 Generator Model

We looked into multiple open-source pre-trained large language models (LLMs) relevant to our project and compiled a list of those we found most suitable:

PHI-2 (Li et al., 2023) This model has 2.7B parameters. When assessed against benchmarks testing common sense, language understanding, and logical reasoning, Phi-2 showcased a nearly state-of-the-art performance among models with less than 13B parameters. The Phi-2 model is best suited for prompts using the QA format, the chat format, and the code format. This model has not been fine-tuned yet, which gives us room to do so on the tasks we need. Moreover, we expect it to be large enough to produce good results while being quicker to train, allowing us to try multiple approaches.

Flan T5-XL (Chung et al., 2022) This model has 2.85B parameters. It uses the T5 model architecture (Raffel et al., 2020) and has been fine-tuned for instructions, achieving better scores in some benchmarks than much larger models.

2.2 Quantization Specialization

We chose quantization because we recognize the significant advantages of reducing the size of our model. A smaller, more efficient model could operate on devices with lower power/energy capacities. By employing quantization, our model can perform effectively without relying on extensive cloud infrastructure, which can often introduce latency and require more energy.

We intend to use the quantization schema presented at (Xiao et al., 2024). This method does not need further training and obtains considerable speedup and memory efficiency while maintaining the performance of the model. We intend to use per-tensor static quantization, using INT8.

3 Data

We present some datasets and benchmarks that will be used to train and evaluate the generator model.

Importantly, the datasets that will be used for the 2nd SFT will be pre-processed, so that answers follow the template `{"explanation": ..., "choice": A}`. This way, the model learns to output answers using the template. During inference, the model output to a prompt can be post-processed *s.t.* only the `"choice"` is returned to the user. We theorize that maintaining the `"explanation"` in the output leverages better the *Chain-of-Thought* capabilities of the model, yielding better results.

3.1 Generator Model

Orca-Math (Mitra et al., 2024) This dataset focuses on grade school math word problems. All the answers in this dataset are generated using Azure GPT4-Turbo. As it is a recent dataset, it gives us some guarantees that the chosen model has not been trained using this data.

MMLU (Hendrycks et al., 2021) This is a benchmark consisting of multiple-choice questions from various branches of knowledge. In particular, it contains questions from college-grade maths, physics, and engineering. Thus, it is a good benchmark to assess the model performance in answering questions from EPFL.

stemQ (Drori et al., 2023) A dataset of 667 questions and solutions from 27 STEM courses across 12 departments in 7 universities.

Open Code Interpreter Dataset (Zheng et al., 2024) This dataset includes programming and coding questions, which are essential to assist users in interpreting and explaining code snippets.

3.2 Quantization Specialization

We intend to use Falcon RefinedWeb (Penedo et al., 2023) to extract calibration samples for the quantization process, as this dataset was used to pre-train the Phi-2 model.

4 Evaluation

In our project, we will apply both a qualitative evaluation method and the automatic evaluation method developed in AlpacaFarm (Dubois et al., 2024) to assess the performance of our chosen model. The qualitative evaluation method will consist of "blind testing" pairwise outputs from our model and the outputs of our chosen baseline model, e.g. Davinci003. We will select and train a group of evaluators from our class. Each response generated by our chatbot will be reviewed by at least two evaluators to ensure a balanced perspective. For each output pair, evaluators will select the preferred output giving us in the end a qualitative win rate over against the baseline model.

The automatic evaluation will use a simulated win-rate system to compare our model's outputs against those from our baseline model. To replace human evaluators, we use simulated annotators created by querying a state-of-the-art language model like GPT-4. These simulated annotators generate preferences by evaluating pairs of outputs for given instructions, i.e. the ones we used for DPO (Overall ranking, correctness, relevance, clarity, and completeness). By evaluating how often our model is preferred over the baseline in various tests, we can quantitatively measure its effectiveness. The use of simulated annotators reduces the time spent and the cost of evaluation compared to hiring human annotators, making it feasible to do evaluations on a larger scale. These two approaches will allow us to identify areas for improvement in our model where the costly and time-consuming human evaluations are thought of as support to the larger scale evaluations from the automatic simulation framework.

5 Ethics

Our goal with this project is to develop a reliable and safe chatbot that EPFL students theoretically could use daily for academic purposes. We are committed to ethical practices, ensuring our model provides high-quality academic responses and does not generate harmful or biased content. To achieve this, we will thoroughly check the model's outputs and update our training methods if needed. We will also respect user privacy by never using personal data without explicit consent. We believe that this approach will help maintain integrity and trust in our solution.

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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

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.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback.

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.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report.

Rabeeh Karimi mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2024. Smoothquant: Accurate and efficient post-training quantization for large language models.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2024. Opencodeinterpreter: Integrating code generation with execution and refinement.