MNLP Project Proposal

Ali Bakly | 383057 | ali.bakly@epfl.ch
Elias Ulf Hörnberg | 384928 | elias.hornberg@epfl.ch
Group: ab-eh-me

1 Introduction

In this project, we aim to develop a chat assistant tailored for EPFL students by finetuning existing open-source large language models (LLMs) with human preference data. This data will be derived from responses generated by ChatGPT to a database of questions commonly used by EPFL staff across various technical curricula. These responses will then be evaluated and ranked by EPFL students to serve as human feedback for finetuning the models. Additionally, we plan to implement quantization techniques to reduce the model size, thereby enhancing inference speed without significantly compromising performance.

2 Model

For our chat assistant, we will utilize pretrained models that we will fine-tune using our preference data. We will select several models of varying sizes (number of parameters) because we may have limited data, and a model that is too small might be too simplistic to capture the complexity of the data.

2.1 Generator Model

To select a base model for fine-tuning, we are considering several promising semi-open source models that have recently been released, such as Llama 3 (AI@Meta, 2024) and Phi-3 (Abdin et al., 2024). We plan to utilize the "small" editions of these models — specifically, Llama-3-8b and Phi-3-3.8b. Should we find that these models do not sufficiently learn from the preference data, we may experiment with a smaller base model, such as Flan T5-Large (Chung et al., 2022), which has 783 million parameters.

To develop our final generator model, we fine-tune the base model using our preference data in accordance with the Direct Preference Optimization (DPO) algorithm (Rafailov et al., 2023). DPO, a relatively recent method, offers a direct approach to teaching the model preferences, contrasting with the widely used Reinforcement Learning from Human Feedback (RLHF) paradigm. By employing DPO, we can bypass both the reward modeling and reinforcement learning phases, directly optimizing the maximum likelihood objective instead:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \log \mathcal{P}_+ - \beta \log \mathcal{P}_-\right),$$
$$\mathcal{P}_+ = \frac{P_{\text{curr}}(Y_+|x^*)}{P_{\text{base}}(Y_+|x^*)}, \quad \mathcal{P}_- = \frac{P_{\text{curr}}(Y_-|x^*)}{P_{\text{base}}(Y_-|x^*)}.$$

In this context, σ represents the sigmoid function, and β is a control parameter. P_{curr} refers to the likelihood from the model currently undergoing fine-tuning, while P_{base} represents the base model. The variables Y_+ and Y_- denote the preferred and rejected data points, respectively, and x^* represents the input instruction tokens. For the purpose of fine-tuning, we will utilize the `DPOTrainer` from the Python library `trl` (von Werra et al., 2020).

2.2 Quantization Specialization

To enhance the practicality and usability of our fine-tuned model, we plan to implement quantization to reduce the model size and improve inference speed. Quantization involves converting the model's floating-point parameters to lower-precision representations, such as int8 or float16, which require less memory and computational resources (Gholami et al., 2021). We will mainly explore post-training quantization (PTQ), and for this purpose we plan to employ already built tools by Hugging Face.

3 Data

The preference data will comprise questions, each paired with two corresponding answers, and an indication of preference for each pair. This data will be generated by us, the students of CS-552: Modern Natural Language Processing at EPFL. Additionally, we will explore alternative data sources if

we find that the student-annotated data is insufficient.

3.1 Generator Model

Each student will create a dataset consisting of 100 preference pairs, with questions supplied by EPFL staff across various technical curricula. We anticipate the complete dataset to contain between $10,000$ and $20,000$ preference pairs. The generation of these pairs will be facilitated by ChatGPT, after which it will be the students' responsibility to fact-check and rank the responses.

For our own set of 200 preference pairs, we plan to experiment with different prompting techniques to generate high-quality responses. We will leverage the capability of Large Language Models to function as Zero-Shot Reasoners, utilizing key phrases such as "Let's think step by step" (Kojima et al., 2023). We also implement the ideas presented in a paper by Google DeepMind, which found that the phrase "Take a deep breath and work on this problem step-by-step" increased performance (Yang et al., 2024). Additionally, we will explore the use of multi-prompting techniques that help contextualize the question, akin to Few-Shot Learning (Brown et al., 2020).

Furthermore, we will investigate whether the incorporation of other publicly available datasets can enhance performance. We have observed that some questions are heavily arithmetic-focused, a domain in which large language models typically struggle. Therefore, external datasets specializing in mathematics or algebra may be necessary. For example, the distilabel-math-preference-dpo dataset (Daniel and Francisco, 2023), focuses on high school mathematics topics such as algebra and calculus. For more advanced mathematics, the preference-data-math-stack-exchange dataset, derived from the H4 Stack Exchange Preferences Dataset (Lambert et al., 2023), could be utilized. If our model requires additional data from other subject areas, such as physics or computer science, the original H4 Stack Exchange Preferences Dataset may prove useful.

4 Evaluation

For the evaluation of the fine-tuned model, we will employ various scoring metrics. While we anticipate using BERTScore (Zhang et al., 2020), BLEU (Papineni et al., 2002), and ROUGE (Lin, 2004), we may consider additional metrics as well. We will also conduct human inspections as a sanity check to ensure the reliability of our results.

4.1 Generator Model

We plan to test the pretrained model both before and after fine-tuning to assess improvements in performance. Additionally, for the manual human inspection, we might compare our model with widely recognized generated models online, such as ChatGPT, to see if ours performs better at specific tasks and to better gauge its performance. If our API budget permits, we could also evaluate ChatGPT using the mentioned scores as another comparison metric. However, our primary focus will be on the relative improvement observed before and after fine-tuning.

4.2 Quantization Specialization

The same scoring system will be used to compare the fine-tuned model before and after quantization. A slight decrease in performance is expected following quantization, but it is difficult to predetermine what an acceptable loss in performance would be. Ideally, the model should still perform better than it did before fine-tuning.

5 Ethics

This generator model is trained on a dataset that is, hopefully, reliable; however, errors may exist as several hundred students are involved in its creation. Such mistakes could lead the model to generate incorrect answers, potentially misleading users who rely on it for assistance. Additionally, the model could be used for cheating, since it is trained on questions from EPFL courses, enabling students to obtain solutions without solving the problems themselves. If misused, this could hinder learning, but if used appropriately, it could enhance it. Moreover, as the questions are provided by the course staff, it is unclear whether these questions are sourced from exams, which could further complicate issues of cheating.

Our fine-tuned model is designed to assist with EPFL course questions, which are typically science-focused, but using it for other purposes might result in less inclusive language. This is because we do not plan to train the model on "helpfulness" or "harmlessness" preference data, such as the HH-RLHF dataset (Bai et al., 2022).

Finally, since the model is not intended for public release, many of these concerns are mitigated.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, 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, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, 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, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, 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.

AI@Meta. 2024. Llama 3 model card.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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.

Vila-Suero Daniel and Aranda Francisco. 2023. Argilla - Open-source framework for data-centric NLP.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. Huggingface h4 stack exchange preference dataset.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

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`.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.