MNLP Progress Report

Arthur Chansel | 324265 | arthur.chansel@epfl.ch
Iris Segard | 310860 | iris.segard@epfl.ch
Gilles de Waha | 330072 | gilles.dewaha-baillonville@epfl.ch
MoreNectarLessPoison

1 Introduction

In this progress report, we will expose the current state of our project: build a real-world LM for education assistance. We have been training a generator model that creates completions aligned to the preference data. First, we will present the dataset. Second, we will discuss the base model selected. Then, we will detail our findings. Finally, we will explain the specialization of the model with Quantization.

2 Dataset

The DPO approach (Rafailov et al., 2023) specifically requires preference data. A data point consists of a prompt and pair of response, where one is "good" and the other is "bad". Hopefully, every students of the MNLP class labeled a tailor made dataset for this project. We use this data to train our model with DPO.

Furthermore, we preprocessed the dataset to ensure the high quality of the data. First, we checked if the overall preference was either response A or B. Then, we checked if the samples were coherent, and we discarded those whose overall preference doesn't appear in any criteria. Indeed, if the overall preference is response A, but no other criteria prefers response A, the sample might not contain sufficient quality. In the end, we discarded 2243 preference pairs, while keeping 24495 pairs before splitting with ratio 80%-20%.

3 Model

The model selection was a challenge, we had to change it multiple times. We first aimed to use Llama 3 (Meta, 8B), until we tried to run it and realized that it would take too much time to train on the dataset. Initially we aimed to freeze an important part of the parameters with LoRa, but this approach is complex, and our teaching assistant advised us to think of an alternative. We switched to Phi-2 Instruct (Microsoft, 3B) but again we were concerned about its size and the fact that it wasn't instruct. Finally, we turned to OpenLM (Apple, 450M-Instruct) which was more affordable. This model uses a tokenizer from Llama 2 (7B). It was a good trade-off between computational cost and efficiency. We decided not to do SFT before DPO since the model was already instruct and trained on a variety of scientific datasets such as Github, StackExchange and ArXiv fig. 1.

Source	Subset	Tokens
RefinedWeb		665 B
RedPajama	Github	59 B
	Books	26 B
	ArXiv	28 B
	Wikipedia	24 B
	StackExchange	20 B
	C4	175 B
PILE		207 B
Dolma	The Stack	411 B
	Reddit	89 B
	PeS2o	70 B
	Project Gutenberg	6 B
	Wikipedia + Wikibooks	4.3 B

Figure 1: Dataset used for pre-training OpenELM.

We use the default loss function of the DPO Trainer. It is a sigmoid loss on the normalized likelihood. (Rafailov et al., 2023)

4 Preliminary Training Results

While training our model on the preference data, the logs of reward metrics are recorded[1] for the training set. We can visualize them with the tool wandb[2].

First, the training loss is recorded in fig. 2.

[1] https://huggingface.co/docs/trl/main/en/dpo_trainer#logging
[2] https://wandb.ai/

Then, the rewards values for chosen and rejected responses are logged in fig. 3 and fig. 4. Specifically, the value displayed corresponds to the mean difference between the log probabilities of the policy model and the reference model

Finally, the rewards accuracies in fig. 5 tell us how often the chosen rewards are higher than the rejected rewards.

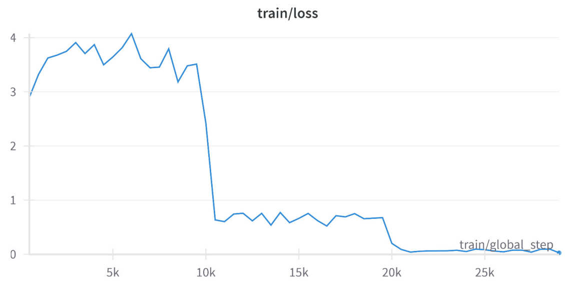

Figure 2: Train loss

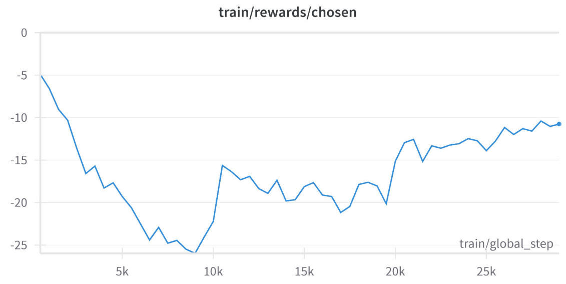

Figure 3: Rewards values for chosen responses

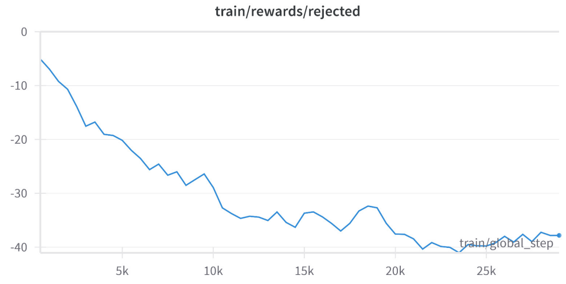

Figure 4: Rewards values for rejected responses

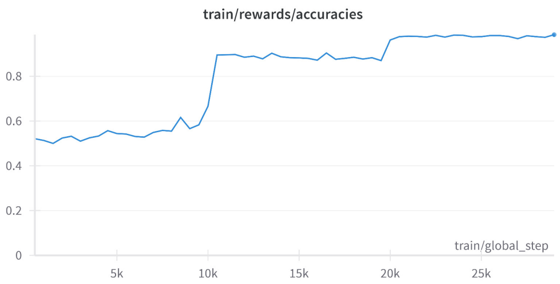

Figure 5: Rewards accuracies on the train set

As anticipated, we observe a reduction in loss and an improvement in accuracy for the training set. We also notice an increase in the reward for the selected answer, and a decrease in the reward for the rejected answer. These outcomes indicate that our model's training was successful.

However, when testing our model with our test set, we obtain an accuracy of **50.5%**. The accuracy measures the percentage of samples for which our model correctly assigns a higher reward to the chosen response. Since the random baseline is 50% and we are only slightly above it, our model performs poorly. The big difference between our train accuracy (98%) and test accuracy (50.5%) indicates that our model severely overfits.

5 Quantization Specialization

For the specialization we intend to quantize our model to int8 from bfloat16. We count on using the quantization library from Hugging Face[3] to implement the LLM.int8() algorithm (Dettmers et al., 2022). For further evaluation, we intend to compare performance between models with the MMLU benchmark (Hendrycks et al., 2021).

References

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale.

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

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.

[3] https://huggingface.co/docs/transformers/en/main_classes/quantization