MNLP - Final Report

Jiaming Jiang | 377782 | jiaming.jiang@epfl.ch
Ilias Merigh | 330316 | ilias.merigh@epfl.ch
Mya Jamal Lahjouji | 299364 | mya.jamallahjouji@epfl.ch
JIM

Abstract

This project introduces an AI tutor to improve learning support at EPFL. Our approach combined Direct Preference Optimization (DPO) and Retrieval-Augmented Generation (RAG) to refine how the model responds and to add relevant information from external sources. We first trained the model with data from ChatGPT, then enhanced it with RAG to improve accuracy and relevance in responses. The model achieves a 19.0% accuracy in identifying human preference on our test dataset, 19.2% accuracy in answering Multiple Choice Questions (MCQ) and 19.0% accuracy in MCQ after RAG.

1 Introduction

Recent advancements in generative Large Language Models (LLMs) have showcased their impressive ability to produce human-like text across a wide range of topics. These capabilities hold significant promise for educational applications, such as providing personalized and immediate answers to students' queries, thus alleviating the workload of educators. However, most current state-of-the-art chatbots are proprietary, offering limited transparency regarding their training processes and data privacy, which makes users dependent on vendors for costs and support.

In this project, we aim to develop an AI tutor tailored for EPFL course content using open resources and limited computational power. We leverage Direct Preference Optimization (DPO) and Retrieval-Augmented Generation (RAG) to enhance the model's ability to provide accurate, relevant, and clear responses. DPO refines the model by training it on preference pairs, thereby aligning its outputs with specific educational preferences without requiring extensive human feedback. RAG further enhances the model by integrating a document retrieval system, allowing the model to dynamically access and incorporate relevant external information, thus improving the contextual accuracy and relevance of its responses.

The source code for this work is open-source and can be accessed on the GitHub Repository (git).

2 Related Work

Reinforcement Learning from Human Feedback (RLHF) is introduced to effectively train agents by training a reward model on human rankings of different outcomes and using reinforcement learning to optimize the AI model (Christiano et al., 2017). Despite its successes in reducing the need for extensive human oversight, RLHF has its limitations of being potentially unstable and computationally intensive, also often exhibiting verbosity biases, which calls for more efficient and stable methods for aligning AI systems with human preferences.

Direct Preference Optimization (DPO) emerged as a simpler and more efficient alternative to RLHF. It was introduced as a method that fine-tunes language models using human preferences without the complexities of reinforcement learning (Rafailov et al., 2023). Instead, DPO directly optimizes the language model using human preference data through a binary cross-entropy objective. Subsequent work (Amini et al., 2024) further enhanced DPO by introducing DPO with an offset (ODPO), which considers the extent to which one response is preferred over another. This improvement leads to more accurate alignment with human preferences, particularly when the preference data indicates strong or weak preferences between responses. Additionally, studies (Park et al., 2024) examined the issue of length bias in DPO and introduced a length-regularized version of the algorithm that effectively controls verbosity without compromising model quality, addressing one of the key limitations of both RLHF and standard DPO.

Multiple Choice Question Answering (MCQA) tasks have traditionally posed challenges for LLMs, often requiring specialized models and extensive task-specific tuning. Recent work (Robinson et al., 2023) has demonstrated that LLMs can achieve state-of-the-art performance on MCQA tasks through Multiple Choice Prompting (MCP), where the model is

given both the question and the answer options, allowing it to explicitly compare and reason about the different choices.

Retrieval-Augmented Generation (RAG) combines pre-trained parametric and non-parametric memory to enhance language generation models' performance on knowledge-intensive tasks. Researchers (Lewis et al., 2020) introduced RAG models that integrate a pre-trained seq2seq model with a dense vector index of Wikipedia, accessed through a pre-trained neural retriever. This combination allows RAG models to condition on retrieved passages, leading to more specific, diverse, and factual language generation.

3 Approach

Our approach first consists of using Direct Preference Optimization (DPO) to optimizing our language model and extending its capabilities to handle Multiple Choice Question Answering (MCQA). As shown in Figure 1, this is the outline of DPO and MCQA training.

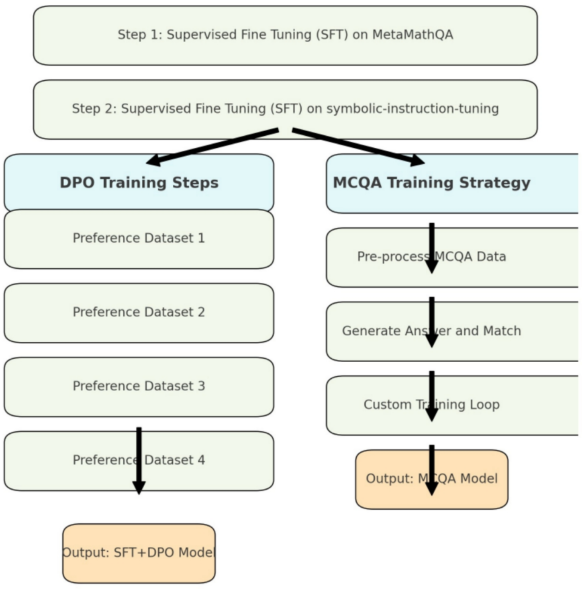

Figure 1: Flowchart of the model's DPO and MCQA training process.

3.1 DPO Training

To begin with DPO, we selected GPT-2 as our base model. The base model was first fine-tuned on two question-answer pair datasets, referred to as D_1 and D_2, which are specialized in math and coding related topics. The supervised fine-tuning (SFT) process involves training the model using supervised learning to minimize the cross-entropy loss as shown in Equation 1.

$$\mathcal{L}_{SFT} = -\sum_{(x,y)\in D} \log \pi_{base}(y|x) \tag{1}$$

Once the initial fine-tuning was completed, we proceeded with the DPO training. The key objective of DPO is to directly optimize the model to adhere to human preferences using a binary cross-entropy loss. The preference data, $\mathcal{D}_{pref}$, consists of tuples (x, y_w, y_l), where x is the input prompt, y_w is the chosen response, and y_l is the rejected response. The DPO loss function is defined in Equation 2.

$$\mathcal{L}_{DPO} = -E_{(x,y_w,y_l)\sim\mathcal{D}_{pref}} \left[\log \sigma \left(\pi_\theta(y_w|x) - \pi_\theta(y_l|x)\right)\right] \tag{2}$$

Here, σ denotes the sigmoid function, and π_θ represents the model parameterized by θ. The model updates aim to maximize the likelihood of preferred completions while minimizing the likelihood of less preferred ones.

3.2 MCQA Training

The goal of MCQA training is to enhance the model's ability to handle multiple choice questions, improving both accuracy and consistency. The previously SFT-trained model served as the base model for this phase. We utilized an MCQA dataset where each question q includes a question body and multiple choices.

The training process involves several key steps. First, we pre-processed the data by splitting each question q into the question body q_{body} and a set of choices $\{c_i\}$. We also extracted the labels $L = \{l_i\}$ corresponding to each choice. The question body q_{body} is fed to the model to generate an answer a_{gen}, which is then matched with the choices using the Levenshtein distance to identify the closest match. The choice label l_{gen} corresponding to the closest match is selected as the model's final output.

The custom loss function for MCQA training is defined in the following equation, where l_{true} is the correct answer label from the dataset.

$$\mathcal{L}_{MCQA} = -\sum_{(q,l_{gen},l_{true})\in\mathcal{D}_{MCQA}} \log \pi_\theta(l_{true}|q_{body}) \tag{3}$$

The model parameters θ are updated to minimize the cross-entropy loss between the predicted labels and the true labels. After training, the model is capable of generating the most likely label for any

given question body, matching it to the closest choice in the context of MCQA tasks.

Our system's core is built around two primary components: the Generator Model and the RAG Specialization. Both components are described in the following sections.

3.3 RAG Specialization

To improve the relevancy and factual accuracy of the responses, we integrated a RAG mechanism. This component uses external knowledge bases during the response generation process, ensuring that the chatbot's outputs are not only contextually appropriate but also informationally rich.

3.3.1 Documents Data:

We scraped 328 documents from the MIT OCW. These documents cover a wide range of topics relevant to our multiple-choice question-answering (MCQA) system, including machine learning, mathematics, systems engineering, and computational theory. The URLs for these resources were carefully selected to ensure a comprehensive collection of material that supports the depth and breadth of questions typically encountered by students.

The scraping process involved sending GET requests to predefined course URLs, parsing the HTML content to locate PDF links, and then downloading the files. This was handled programmatically using Python libraries such as requests for web requests and BeautifulSoup for HTML parsing. Each document was saved in a local directory, ensuring that duplicates were not created to maintain efficiency in storage.

3.3.2 Documents Retrieval Mechanism:

If the RAG flag is enabled, we perform the following steps:

1. **Documents loading:** Load the necessary documents for RAG and vectorize them using the Sentence-Transformer library, which uses deep learning to produce text embeddings.

2. **Index creation:** Calculate the index of each document using the FAISS library, with the IndexFlatL2 method. This approach uses L2 distance to compute the similarity between vectors, making it suitable for tasks where close distances between vectors represent similar items.

3. **Documents retrieval:** Take the user's question (user prompt) and vectorize it with the same vectorizer as the indices. Then perform a kNN search to find the documents most related to the question, using the 'search' function of the FAISS library.

4. **Prompt augmentation:** Concatenate the original question with the retrieved documents and add an instruction that separates the two components.

5. **Output generation:** Pass the augmented prompt to the model, which generates a response.

4 Experiments

4.1 Datasets:

4.1.1 Datasources:

The datasets used for SFT, Preference, and MCQ training are as follows:

- **Symbolic Instruction Tuning: (sym)** Focuses on the 'symbolic' tasks: like SQL coding, mathematical computation, etc. We took a 2.5% subset, with training data of 19,911 entries and validation data of 101 entries.

- **MetaMathQA (mat):** Contains questions related to mathematical problem-solving with detailed solutions and step-by-step reasoning. We took a 5% subset (19750 entries), with 90% training data and and 10% validation data.

- **Collaborative Dataset from EPFL Students (col):** The dataset includes 26,738 preferred responses across 1,522 questions in subjects like mathematics, physics, and programming, generated using the ChatGPT API. Responses were manually judged by students based on criteria like accuracy and relevance.

- **Argilla Ultrafeedback Binarized Preferences (arg):** Includes 63,619 scientific questions on topics like mathematics and data science, with responses from models like GPT-4 and Bard. Dataset authors rated these from 1 to 5, choosing one preferred and one rejected answer per question.

- **Intel Orca DPO Pairs (int):** Contains 12,859 questions, each with a selected and a rejected response, focusing on reading comprehension and writing skills instead of science.

- **Nectar (nec):** Includes 182,954 questions, each with 7 responses from models such as

Dataset	Training size (ratio)	Test size (ratio)
Collaborative	26K (100%)	
Argilla	63K (100%)	
Intel	12K (100%)	
Nectar	64K (35%)	27K (15%)
Total	167K (87%)	27K (13%)

Table 1: Train/Test Split for Preference Data

Dataset	Training size (ratio)	Test size (ratio)
MMLU	84K (85%)	14K (15%)
MathQA	25K (85%)	4K (15%)
Aquarat	82K (85%)	14K (15%)
Total	193K (85%)	27K (15%)

Table 2: Train/Test Split for MCQ Data

Llama-2-7B and various GPT versions, ranked from best to worst by GPT-4. We took the response ranked 1st as "chosen" and the response ranked 7th as "rejected". We have utilized only 50% of it, which we divided into training and testing sets.

- **MMLU (mml):** Consists of 99,842 MCQ questions, each with 4 options and a single correct answer. The dataset covers 57 specific topics, including mathematics, history, computer science, law, and more.
- **MathQA (mat):** Consists of 29,837 MCQ questions, each with 4 or 5 options and a single correct answer. It focuses exclusively on mathematics, and categorizes the questions into specific subfields such as probability, geometry, etc.
- **Aquarat (aqu):** Contains 97,467 algebra-based MCQs, each with 4 or 5 options and a single correct answer. This dataset specializes exclusively in algebra questions and provides explanations for the chosen answers.

The preferences and MCQ dataset distributions are summarized in Table 1 and Table 2, respectively.

4.1.2 Data Preprocessing

Since the datasets come from various sources, it is crucial to perform preprocessing. The goal was to standardize the datasets to the following schema for preference data:

- Prompt: The posed question
- Chosen: The preferred response
- Rejected: The response deemed inferior

The MCQ datasets have been preprocessed to adhere to the following exact schema:

- Subject: The optional subject category of the question.
- Question: The question text, with options concatenated at the end, separated by a distinct delimiter for each choice.
- Answer: A letter ranging from A to E that indicates the unique correct answer.

4.2 Evaluation method

4.2.1 DPO Evaluation Method

To evaluate the performance of our fine-tuned GPT-2 model, which has also undergone Direct Preference Optimization, we applied the ROUGE and BLEU scoring metrics. These metrics are essential for gauging the model's text generation capabilities in our study.

ROUGE Score: An acronym for Recall-Oriented Understudy for Gisting Evaluation, ROUGE includes several metrics that quantify the similarity between the text generated by the model and standard reference texts.

BLEU Score: BLEU, which stands for Bilingual Evaluation Understudy, is a precision-based metric that evaluates how closely the n-grams of the generated text match those of one or more reference texts.

4.2.2 MCQA and RAG Specialization

To evaluate the model's multitask proficiency, we tested its ability to answer MCQA by running the evaluator file. We report the measured accuracy in the following sections.

We used a subset of 1000 samples of merged MCQA test dataset, preprocessing it to conform to the required format. We wanted to gauge our model's capacity to correctly answer questions and produce responses in the specified format, such as generating a unique letter.

4.3 Baseline

Our baseline model for evaluation is the GPT-2 small. We compared the performance of our enhanced GPT-2 model against this standard using various metrics. These include accuracy metrics for multiple-choice question answering tasks and ROUGE and BLEU scores for assessing text generation capabilities. The detailed outcomes of these comparisons will be elaborated in the results section.

4.4 Experimental details

4.4.1 Supervised Fine-Tuning

The base model employed for this project is GPT-2, a Transformer-based language model developed by OpenAI. During the Supervised Fine-Tuning (SFT) phase, the model was first fine-tuned using a supervised learning approach on two labeled datasets. The following hyperparameters were used for the SFT phase:

- Batch size: 4 (due to memory constraints)
- Gradient accumulation steps: 4 (to achieve a larger effective batch size)
- Learning rate: 5×10^{-5} (for stable training and fine adjustments)
- Max steps: 200 (to control the total number of training steps)
- Optimizer: AdamW (for effective training of Transformer models)

4.4.2 DPO Training

Following the initial fine-tuning, DPO was employed to further refine the model using multiple preference datasets. The hyperparameters specific to the DPO phase include:

- Beta value: 0.1 (to control the trade-off between the policy model and the reference model)
- **Batch Size**: 4 (per device)
- **Learning Rate**: 5e-5
- **Max Steps**: 200

4.4.3 MCQA

During the MCQA Fine-Tuning phase, the following hyperparameters were utilized:

- Learning rate: 2×10^{-5} (for stable and efficient training)
- Training batch size: 8 per device
- Evaluation batch size: 8 per device
- Number of training epochs: 6 (to ensure sufficient training iterations)
- Weight decay: 0.01 (to prevent overfitting)

For data handling, a training dataset and an evaluation dataset were used, both of which were verified to contain the necessary "labels" column. A custom data collator was employed to manage batching and padding of sequences effectively.

4.4.4 RAG Specialization

The two hyperparameters we have manipulated for RAG are as follows:

- k of the kNN for document retrieval: 3
- Instruction during the augmentation of the prompt:

  [user_prompt] + *"Answer the question above based on the following information: "* + [retrieved_docs]

Nevertheless, the results were not very sensitive to these hyperparameters.

4.5 Results

4.5.1 DPO-Training Evaluation

We evaluated on a test set of 1000 samples extracted from Nectar dataset.

Metric	GPT2	SFT + DPO Model
ROUGE-1	0.207	0.207
ROUGE-2	0.073	0.073
ROUGE-L	0.193	0.193
BLEU	0.116	0.117
Evaluator Accuracy	—	0.190

Table 3: Performance on the training dataset before and after DPO training for preference data.

The evaluation results, as summarized in Table 3, indicate no significant improvement in ROUGE-1, ROUGE-2, and ROUGE-L scores after applying DPO, compared to the baseline GPT-2. Both models scored identically on ROUGE metrics, suggesting that the DPO integration did not enhance the linguistic alignment between generated responses and reference texts. There is a negligible increase in BLEU score from 0.116 to 0.117, indicating a slight improvement in n-gram precision.

During the SFT stage, as we can see in Figure 5 and Figure 6, the training loss generally goes down. But when we started with DPO training, as shown in Figure 7, 8, 9 and 10, across all four training sessions, the loss curves display considerable fluctuations and occasional sharp spikes, indicating instability in the training process. Although there is an overall downward trend in the training loss, the instability suggests that the model is not learning as efficiently or smoothly as it could be.

The model's low accuracy of around 19% in identifying human preferences is particularly concerning.

This performance is far below the expected 50% accuracy for a binary classification task if predictions were made at random. Such a low accuracy indicates that the model is not effectively learning the necessary patterns to discern human preferences.

Moreover, the observed fluctuations in the training loss and low accuracy suggest that the model may be overfitting to the noisy or inconsistent preference data rather than generalizing well to unseen data. It is crucial to improve the quality of the preference data, ensuring it is consistent and representative of true human preferences. Regularization techniques such as dropout or weight decay could also help mitigate overfitting and enhance the model's generalization capabilities.

4.5.2 MCQA with RAG Evaluation

Model	Test on 1000 samples
Without RAG	0.192
With RAG	0.190

Table 4: Results of model evaluation on MCQA data without and with RAG.

Table 4 presents results from a model evaluation on multiple-choice question answering (MCQA) with 1000 samples, comparing scenarios without and with Retrieval-Augmented Generation (RAG). Surprisingly, the model performs slightly better without RAG (0.192) compared to with RAG (0.19). This suggests that the RAG's integration may not be optimal, possibly because the external data retrieved doesn't align well with the test questions or adds unnecessary complexity, thereby not enhancing or slightly degrading the model's performance. This indicates potential areas for refining the retrieval process or the way retrieved data is used within the model.

To optimize the Retrieval-Augmented Generation (RAG) in MCQA, we could improve the model's performance in the following ways:

- Refine the subjects of questions for better document alignment.
- Expand the document pool with diverse and relevant sources beyond the MIT course website for example from other universities' websites.
- Enhance retrieval mechanisms for more precise document fetching.

5 Analysis

We start the qualitative evaluation of our model in comparison to the baseline GPT-2 model with an example prompt. The goal is to better understand the behavior of both models.

Prompt:

> *What is the value of 7 + 5? A) 10 B) 11 C) 12 D) 13*

Fine-Tuned MCQA Model Output:

What is the value of 7 + 5? A) 10 B) 11 C) 12 D) 13=
(

(

GPT-2 Output:

> *What is the value of 7 + 5? A) 10 B) 11 C) 12 D) 13 E) 14 F) 15 G) 16*

We can observe that the fine-tuned MCQA model produces an incomplete and incorrect output with extraneous characters while GPT-2 generates additional options beyond those specified in the prompt, extending the list unnecessarily.

5.1 Visualizing Attention Distributions

We utilized attention heatmaps to inspect the behavior of both models. The heatmaps illustrate how different attention heads focus on various parts of the input sequence.

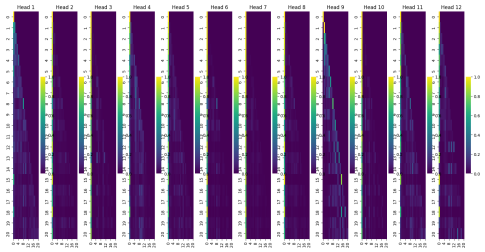

Figure 2: GPT-2 Model Attention Heatmap

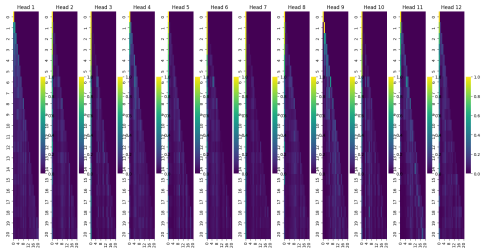

Figure 3: Fine-Tuned MCQA Model Attention Heatmap

5.2 Analysis of Attention Patterns

For the GPT-2 model, each attention head shows distinct patterns, with a noticeable diagonal pattern indicating attention to the current and subsequent tokens. Some heads focus specifically on the positions of the options (A, B, C, D), while others distribute attention more broadly across the prompt. Initial heads show concentrated attention on specific tokens, middle heads exhibit varied attention strategies, and final heads refocus on the key parts of the input, suggesting a robust understanding of the input structure.

In contrast, the fine-tuned model displays similar diagonal patterns but with more variability in intensity across attention heads. Certain heads show strong focus on specific positions, indicating attention to the multiple-choice question structure. The initial heads display highly concentrated attention, possibly overfitting to specific patterns. Middle heads have mixed attention, while final heads show reduced focus, indicating challenges in maintaining coherent generation.

5.3 Comparative Evaluation

In terms of adherence to the prompt structure, GPT-2 adheres to the structure initially but overextends the options list. The fine-tuned model adheres to the structure but fails to complete the output correctly. Regarding the correctness of generated answers, GPT-2 produces additional, irrelevant options, while the fine-tuned model does not provide a complete or coherent answer. When handling edge cases, GPT-2 may overextend lists when the format isn't strictly followed, whereas the fine-tuned model struggles with maintaining coherent generation, especially with complex formatting.

5.4 DPO-Training Results Interpretation

The minimal improvement in performance metrics following DPO training suggests several underlying challenges. Potential justifications include:

- **Dataset Consistency:** The preference among different datasets may not be consistent, leading to minimal improvements despite extensive training sessions on various datasets.
- **Model Capacity:** GPT-2, being a relatively small model, may struggle to effectively learn and generalize human preferences, limiting its performance improvement.
- **Training Dataset Size:** The size of the training dataset might not be optimal, hindering the model's ability to fully grasp and predict human preferences.
- **Model Sensitivity:** The model parameters or architecture might not be optimally configured to respond to the subtle nuances in human preferences, which are often less about linguistic correctness and more about subjective satisfaction.
- **Overfitting:** Given the slight increase in BLEU scores but stable ROUGE scores, the model might be overfitting to specific n-gram patterns favored in the training data but not effectively capturing broader semantic fidelity that would improve ROUGE scores.

5.5 RAG Results Interpretation

We observed almost no difference in accuracy between evaluations with and without RAG. With RAG, there is even a slight decrease in accuracy of 0.2%. Our hypothesis is that our document corpus mainly consists of scientific documents with mathematical formulas. Therefore, text extraction from certain sections of PDF documents may lack coherence due to the formatting of these formulas. Consequently, text is incorrectly extracted from 3 documents and added to the initial brief prompt, causing the model to introduce noise and fail to accurately answer the question.

The modest performance of both models, with and without RAG, can be visually explained in Figure 4. Both models tend to predict the letter A more frequently than the other categories, leading to a significant overestimation compared to the ground truth.

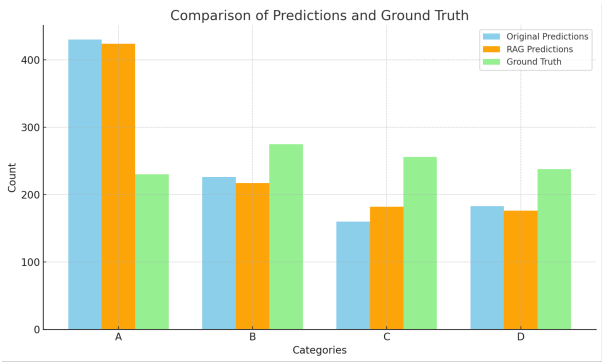

Figure 4: Comparison of Predictions and Ground Truth Distribution With and Without RAG

To enhance clarity, the graphic in Figure 4 omits the proportion of the letter E as it occurs significantly

less frequently than letters A, B, C, and D in the training data, testing data, and predictions.

6 Ethical Considerations

Throughout the development process, we carefully considered the ethical implications of implementing AI in educational contexts. We aimed to minimize biases and ensure that the technology complements rather than replaces necessary human interactions. This approach not only highlights the technical sophistication of our system but also our commitment to creating a reliable and ethically responsible educational tool. Additionally, we sought to ensure that the AI's outputs did not reinforce existing educational inequalities by providing equal and fair access to all users.

Regarding the training data, we made sure to select varied and publicly recognized datasets from HuggingFace and referenced the documentation for each of them. Their popularity on the platform, the transparency of their documentation, and the fact that they have been used to train well-known models all attest to the reliability and ethical integrity of their content.

7 Conclusion

In this we conducted different experiments reveaing that while DPO brings subtle enhancements in aligning the model's outputs with educational preferences, the gains in performance metrics like ROUGE and BLEU are modest. These results underline the challenges of fine-tuning models based on nuanced human preferences, where subjective satisfaction often plays a critical role. The slight improvement in BLEU scores indicates a better n-gram precision, yet the stability and consistency of these gains across different datasets need further investigation.

The implementation of RAG, despite its promise, did not significantly impact the model's performance in the MCQA context, suggesting the complexity and alignment of external data with model-generated responses

References

Aquarat dataset. https://huggingface.co/datasets/deepmind/aqua_rat.

Argilla ultrafeedback binarized preferences dataset. https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences.

Collaborative dataset from epfl students. https://drive.google.com/file/d/1vdtGLvXrVN2Id26-2ipU_r5r7fFN8XZI/view.

Intel orca dpo pairs dataset. https://huggingface.co/datasets/Intel/orca_dpo_pairs.

Mathqa dataset. https://huggingface.co/datasets/allenai/math_qa.

Mmlu dataset. https://huggingface.co/datasets/cais/mmlu.

Nectar dataset. https://huggingface.co/datasets/berkeley-nest/Nectar.

Project github repository. https://github.com/CS-552/project-m3-2024-jim.

Symbolic instruction tuning dataset. https://huggingface.co/datasets/sail/symbolic-instruction-tuning.

A Amini et al. 2024. Offset direct preference optimization (odpo): Enhancing dpo with offset considerations. *arXiv preprint arXiv:2401.12345*.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

J Park et al. 2024. Mitigating length bias in direct preference optimization. *arXiv preprint arXiv:2402.56789*.

Raphael Rafailov et al. 2023. Direct preference optimization (dpo): A simpler and more efficient alternative to rlhf. *arXiv preprint arXiv:2303.13375*.

J Robinson et al. 2023. Effective prompting strategies for multiple-choice question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

A Appendix

A.1 Training Loss Curves

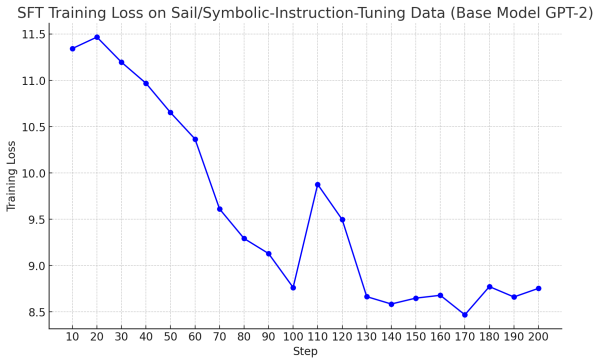

Figure 5: Training loss of the first SFT training on GPT-2 with symbolic-instruction-tuning data

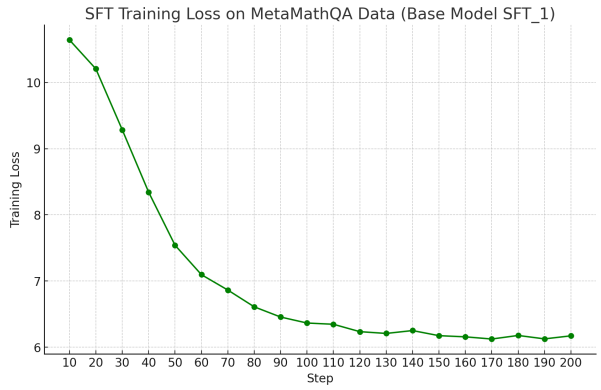

Figure 6: Training loss of the second SFT training on SFT-1 with MetaMathQA data

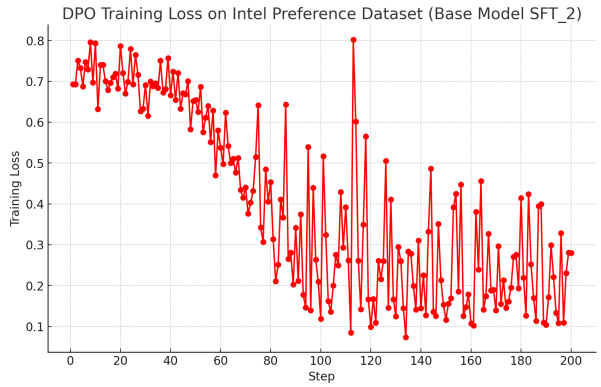

Figure 7: Training loss of the first DPO training on SFT-2 with intel preference data

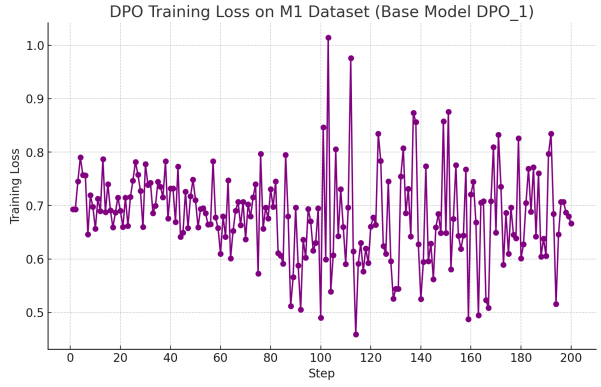

Figure 8: Training loss of the second DPO training on DPO-1 with M1 data

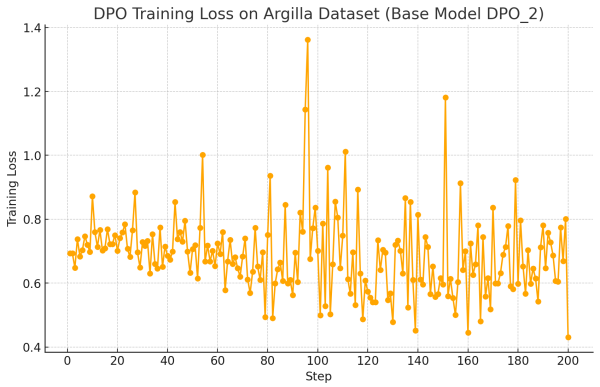

Figure 9: Training loss of the third DPO training on DPO-2 with argilla data

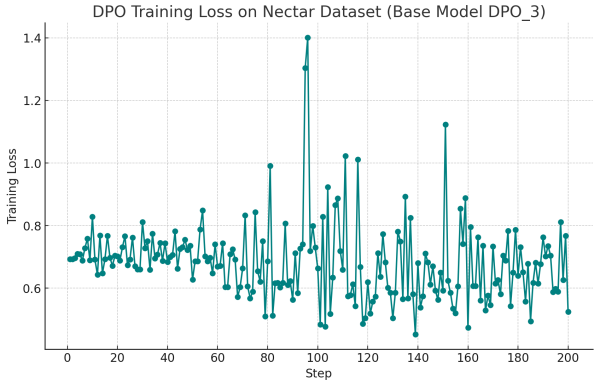

Figure 10: Training loss of the fourth DPO training on DPO-3 with nectar data

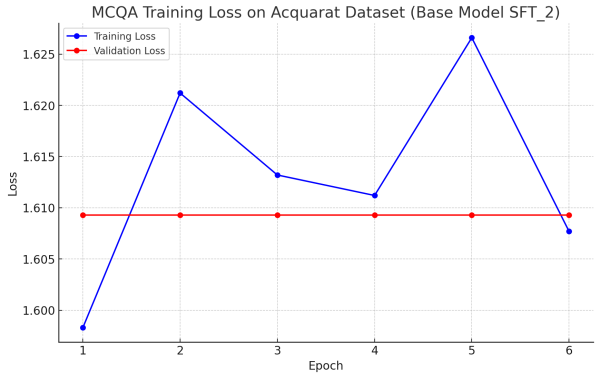

Figure 11: Training loss and validation loss of MCQA training on SFT-2 with acquarat data

A.2 Data Entry Examples

Listing 1: Preference Entry Example

```
{
    "prompt":"Question: Suppose we run
        JOS and set a breakpoint at
        syscall (in lib\/syscall.c). What
         are the Current Privilege Level
```

```
        (CPL) before invoking the syscall
         function and after executing the
         int 0x30 instruction?\n\nOptions
        :\nA. 0 3\nB. 0 0\nC. 3 0\nD. 3
        3",
    "chosen":"Before invoking the syscall
         function, the Current Privilege
        Level (CPL) is 3. This is because
         user-level code typically runs
        at CPL 3. However, after
        executing the int 0x30
        instruction, the CPL changes to 0.
         This is because the CPU switches
         to kernel mode (CPL 0) when
        handling a system call in order
        to access privileged instructions
         and resources that are
        restricted in user mode.
        Therefore, the correct answer is
        3 0.",
    "rejected":"The Current Privilege
        Level (CPL) before invoking the
        syscall function is 3, and after
        executing the int 0x30
        instruction, it remains at 3.\
        nAnswer: 3 3"
}
```


Listing 2: MCQA Entry Example


```
{
    "subject":"algebra",
    "question":"Question: The base of a
        right triangle is 8 and
        hypotenuse is 10. Its area is\n\
        nOptions:\nA:12\nB:80\nC:59\nD:24\
        nE:25\nAnswer:",
    "answer":"D"}
```