GalactiTA: AI-Driven Solutions for Scientific Question Answering

Zied Masmoudi | 330275 | zied.masmoudi@epfl.ch
Jakhongir Saydaliev | 369355 | jakhongir.saydaliev@epfl.ch
Mohamed Amine Ben Ahmed | 300371 | mohamed.benahmed@epfl.ch
NLP-Squad

June 14, 2024

Abstract

Since the revolutionary innovation of Large Language Models (LLMs), the demand for more domain-specific models has surged and EPFL is no exception, seeking an LLM tailored to its unique requirements. In this report, we introduce GalactiTA, a digital teaching assistant (TA) specifically designed to address Multiple Choice Questions in science, technology, engineering, and mathematics (STEM). We began by collecting various Q&A datasets which were then processed to match specific formats. We fine-tuned Galactica-1.3B model for question answering, followed by DPO. Finally, the resulting model was further tuned to respond in a RAG setting, integrating external knowledge in its response. The RAG-tuning alone already improves over the baseline by up to 11.52% and we believe that this model could help students with doubts about their unanswered questions as well as support TAs in responding to challenging students' questions.

1 Introduction

Smaller, open large language models (LLMs) have significantly improved and are now widely applied, including in education. Models like ChatGPT can assist with a variety of topics, but they often produce inaccurate or incoherent reasoning in technical domains. For instance, while they handle basic math and logic problems well, they struggle with complex reasoning. Despite their impressive benchmark performance, LLMs frequently fail to follow user instructions and need better alignment with user intent.

This work studies the following question – *How do we adapt pre-trained LLMs for scientific multiple choice question answering task?*

Inspired by Tunstall et al. (2023), we follow a similar pipeline to align a large language model (LLM) for generating responses to scientific questions. We then enhance the LLM's knowledge using Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) in the Multiple Choice Question Answering (MCQA) format.

We start with the Galactica-1.3B model (Taylor et al., 2022), training it using Supervised Fine-Tuning (SFT) on scientific QA datasets such as ScienceQA (Lu et al., 2022) and SciQ (Welbl et al., 2017). We then utilize ChatGPT to collect preference data primarily based on a question set provided by EPFL and apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to this data. Next, we enrich an MCQA dataset with additional documents from our scientific document collection and generate Chain-of-Thought (CoT) answers using ChatGPT. Finally, we apply SFT to this data, training the LLM to consider the documents in its responses in a RAG setting. Experiments show that RAG tuning effectively improves accuracy by a noticeable 11.52% increase on our collected EPFL dataset, where context-specific retrieval benefits accuracy.

2 Related Work

The development of large language models (LLMs) for scientific applications has seen considerable advancements, with various approaches contributing unique strengths in handling scientific knowledge.

One of the foundational models inspiring our approach is Galactica (Taylor et al., 2022). Trained on diverse scientific resources, Galactica excels in scientific reasoning, LaTeX handling, and benchmarks like PubMedQA and MedMCQA, showcasing its superiority in the scientific domain with state-of-the-art performance.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) transforms the reinforcement learning objective into a classification problem. This method simplifies the alignment process by eliminating the instability associated with traditional RLHF. DPO's success in tasks like summarization and dialogue showcases its potential for improving model alignment in various applications, including question answering.

Building on the idea of enhanced model alignment with user intents, ZEPHYR-7B (Tunstall et al., 2023) leverages distilled supervised fine-tuning (dSFT) and distilled direct preference optimization (dDPO).

To effectively incorporate external knowledge, Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) combines pre-trained parametric and non-parametric memory, outperforming purely parametric models on knowledge-intensive tasks. Using a dense vector index of Wikipedia, RAG enhances factual and specific responses, aligning with our goal of providing accurate, up-to-date information in educational contexts.

Further refining the concept of retrieval-augmented methods, Retrieval-Augmented Fine Tuning (RAFT) (Zhang et al., 2024) was introduced. RAFT combines the methods of SFT and RAG, improving language model performance in domain-specific RAG tasks by training models to cite relevant sequences and ignore distractors, enhancing reasoning and accuracy.

3 Approach

This section provides a detailed overview of our data collection process and the system architecture. The general workflow is illustrated in Figure 1. We begin with a pre-trained model which is fine-tuned for question-answering task followed by a preference alignment, DPO. Lastly, we use RAG to supplement our model's knowledge with information from external sources, where the model is tuned to respond in an MCQA setting.

3.1 Data collection

3.1.1 Instruction data

In the first step, we utilized common scientific question-answering datasets (such as ScienceQA, Sciq, etc...) to create our instruction dataset. It was crucial to include an explanation of the answer to help the model understand the reasoning behind it. For the datasets that provided additional explanations for the questions, we used these explanations as the reasoning that led to the correct answer. We formatted the dataset as follows:

```
prompt: {question}
response: {explanation}. So the answer
is {answer}
```

For other datasets that only provided the correct answer without any additional explanation, we used ChatGPT to syntactically create a CoT answer. Given a question and its corresponding answer, we prompted ChatGPT to generate a CoT answer as follows:

```
Question: {question}
Correct answer: {answer}
Provide an explanation that leads to
choosing the provided correct answer to
the above question.
```

3.1.2 Preference data

The primary source for preference dataset was data produced by our peers at EPFL and ourselves, which comprised of pairs of good and bad responses (referred to as `chosen` and `rejected` in subsequent sections) to EPFL questions. These questions were formatted to suit the DPO alignment data format. We received a set of questions from EPFL courses to generate these samples. To create high quality responses we used prompts such as "Let's think step by step" (Wei et al., 2023) and prompted ChatGPT as illustrated in Figure 2

We applied a similar approach to other datasets to obtain pairs of `chosen` and `rejected` responses. Additionally, we utilized human preference datasets like Stack Exchange, which already contain responses with corresponding scores. For these datasets, we selected the highest scoring response as `chosen` and one of the remaining, at random, as `rejected`. We opted for random selection instead of selecting the lowest-scoring response to encourage diversity and make the DPO objective more challenging.

3.2 RAG data

We collected a set of scientific documents, used in a RAG setting, to increase our model's knowledge (Lewis et al., 2021). We also created a MCQA training dataset with added context and CoT answers to teach the model how to generate responses in a RAG setting.

Documents We collected 5GB of course books in areas such as Computer Science, Artificial Intelligence, Physics, and corresponding scientific papers from arXiv. We then split them into chunks to be used in prompts.

Training data generation We used a set of multiple-choice questions (Q_i) with their corresponding answers (A_i). For each question, we retrieved the top three documents (D_{i1}, D_{i2}, D_{i3}) from our document collection. Inspired by (Zhang et al., 2024), we used ChatGPT to generate a CoT answer (CoT_i) for each question:

$$Q_i + D_{i1} + D_{i2} + D_{i3} + A_i \rightarrow CoT_i$$

We then use the CoT answers (reason) to train our model to respond in RAG settings.

3.3 System architecture

3.3.1 Supervised Fine Tuning

Starting with the base model, we first need to train it to respond to a question. To do this, we concatenate the `prompt` and `response` fields of our instruction dataset and train the model to generate the instance output in a standard supervised way.

3.3.2 Direct Preference Optimization

The fine-tuned model is then aligned with preference data using a preference optimization technique so called DPO (Rafailov et al., 2023). Instead of training a separate reward model like PPO (Schulman et al., 2017), DPO aligns the model's response with the `chosen` response (y_+) over the `rejected` response (y_-) for the given prompt (x) directly through the reformulation of the policy objective as

$$L_{DPO} = -\log \sigma \left(\hat{r}(x, y_+) - \hat{r}(x, y_-) \right) \qquad (1)$$

where $\hat{r}(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$ is the reward implicitly defined by the language model (π) over the reference (SFT) model (π_{ref}) and β is a hyperparameter that controls how much to weight the preference of the reference model.

During the training procedure, we iterate through each triplet (x, y_+, y_-), compute the reward function ($\hat{r}$) for (x, y_+) and for (x, y_-) and minimize the objective given in Equation 1.

3.4 Retrieval Augmented Generation

To improve the model's capability of generating answers using the documents provided in the prompt, we fine-tune our model, using SFT, on the dataset mentioned in 3.2. The training data generation involved retrieving the top 3 contexts for each question, presenting ChatGPT a question, the retrieved contexts and answer, then instructing it to form a CoT response. Figure 3 shows a high-level design steps of our RAG-tuning phase.

4 Experiments

4.1 Data

We used various datasets for the different stages of the training and testing pipeline: SFT, DPO, RAG, and MCQA; See Appendix A for examples.

SFT: We considered 4 datasets:

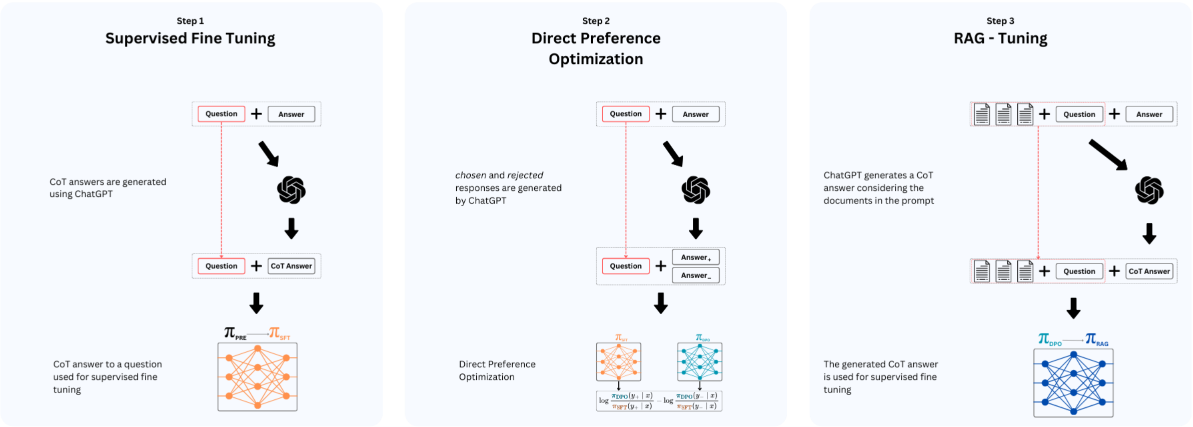
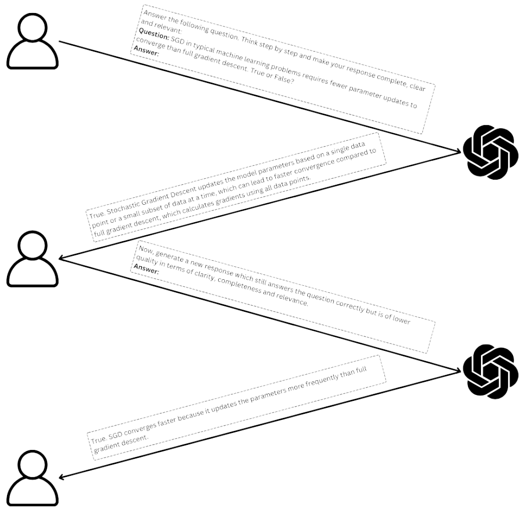

Step 1
Supervised Fine Tuning
Question
Answer
CoT answers are generated using ChatGPT
Question
CoT Answer
$\pi_{PRE} \rightarrow \pi_{SFT}$
CoT answer to a question used for supervised fine tuning
Step 2
Direct Preference Optimization
Question
Answer
chosen and *rejected* responses are generated by ChatGPT
Question
Answer+
Answer_
Direct Preference Optimization
π_{SFT}
π_{DPO}
$\log \frac{\pi_{DPO}(y_+ \mid x)}{\pi_{SFT}(y_+ \mid x)} - \log \frac{\pi_{DPO}(y_- \mid x)}{\pi_{SFT}(y_- \mid x)}$
Step 3
RAG - Tuning
Question
Answer
ChatGPT generates a CoT answer considering the documents in the prompt
Question
CoT Answer
$\pi_{DPO} \rightarrow \pi_{RAG}$
The generated CoT answer is used for supervised fine tuning

Figure 1: The workflow of our pipeline, that consists of SFT, DPO and RAG-tuning.



Answer the following question. Think step by step and make your response complete, clear and relevant:
Question: SGD in typical machine learning problems requires fewer parameter updates to converge than full gradient descent. True or False?
Answer:

True. Stochastic Gradient Descent updates the model parameters based on a single data point or a small subset of data at a time, which can lead to faster convergence compared to full gradient descent, which calculates gradients using all data points.

Now, generate a new response which still answers the question correctly but is of lower quality in terms of clarity, completeness and relevance.
Answer:

True. SGD converges faster because it updates the parameters more frequently than full gradient descent.

Figure 2: Interaction with ChatGPT to collect preference data.

- **ARC** (Clark et al., 2018) dataset which contains 7787 genuine grade-school level, MCQ science questions, assembled to encourage research in advanced question-answering.
- **ScienceQA** (Lu et al., 2022) collected from elementary and high school science curricula on subjects from natural science, language science, and social science, and contains 21,208 multimodal multiple-choice science questions.
- **SciQ** (Welbl et al., 2017) The SciQ dataset contains 13,679 multiple-choice science exam questions on Physics, Chemistry, and Biology, among others, each with 4 answer options. Most questions include a paragraph of supporting evidence for the correct answer.
- **Stack Exchange** (Lambert et al., 2023) This dataset includes questions from an online forum, paired with scored answers based on upvotes.[*]

For each dataset we kept only relevant questions in the domains of Computer Science, AI, Physics, etc. All datasets consist of ~43k data points in total, have been preprocessed as explained in 3.1.1 and have been formatted to match the instruction fine-tuning dataset structure as:

```
prompt: {prompt}
response: {response}
```

DPO: The preference datasets are preprocessed as explained in 3.1.2 and formatted as follows:

```
prompt: {prompt}
chosen: {good_response}
rejected: {bad_response}
```

We categorized them into two groups: HUMAN-ANNOTATED and AI-ANNOTATED:

For AI-ANNOTATED category, we considered the EPFL, ARC and MMLU (Hendrycks et al., 2021) datasets. EPFL dataset has been collected by the students prompting ChatGPT, while MMLU is specifically designed to assess the scientific knowledge of LLMs in MCQA setting.

For HUMAN-ANNOTATED category, we considered the SHP (Ethayarajh et al., 2022) and Stack Exchange datasets that were collected from Reddit and Stack Exchange platforms respectively[†]. For both datasets we filtered only the questions on relevant subjects, e.g. CS, physics, math, etc.

RAG: We considered 4 datasets for training and testing: ARC, MMLU, SciQ and EPFL, that we processed as explained in 3.2 and formatted as:

```
instruction: {instruction}
contexts: {topk_contexts}
question: {question_and_options}
reason: {CoT_response}
answer: {single_letter_answer}
```

MCQA: From our RAG data, we used only the `question` and `answer` fields to tune our model with MCQA. This allows fairly evaluating our RAG-tuning step.

We carry out 4 types of training: SFT, DPO, RAG and MCQA with the corresponding datasets mentioned above.

4.2 Evaluation method

We evaluated our model using 3 automatic, quantitative metrics based on accuracy scores[‡]. These decoding methods were inspired by Hugging-Face (2023) and are illustrated in Figure 4.

1. Token Distribution Method: The model receives a question and the distribution of the next token among A, B, C, and D is examined. The letter with the highest score was selected as the prediction.

[*] We only used the answers with the highest score for SFT.

[†] It is important to note that both the ARC and Stack Exchange datasets were divided into **separate** subsets for DPO and SFT training to ensure that each training phase benefits from the specific strengths of the datasets while maintaining consistency in the content used across both methodologies.

[‡] Defined as the proportion of predicted letters matching the true correct letters

2. Greedy Decoding Method: The model receives a question and uses greedy decoding to predict the next token from the entire vocabulary, rather than limiting the choice to A, B, C, D. The token with the highest probability is chosen as the prediction.

3. Option Probability Sum Method: Given a question, we calculate the probability of each option sequence (letter and its description) by summing the probabilities of its tokens. The letter corresponding to the option sequence with the highest total probability is selected as the answer.

4.3 Baselines

The following baselines are considered:

- **GalactiTA**$_{DPO}$**:** Galactica-1.3B model trained on SFT followed by DPO.[§]

- **GalactiTA**$_{MCQA}$**:** GalactiTA$_{DPO}$ model followed by MCQA-tuning. [¶]

- **GalactiTA**$_{RAG}$**:** GalactiTA$_{DPO}$ model followed by RAG-tuning. [¶]

- **TinyLlama** A chat model fine-tuned on top of TinyLlama 1.1B model with a similar approach to ours, i.e. SFT followed by DPO. For a fair comparison, we also apply MCQA-tuning on it: TinyLlama$_{MCQA}$.[¶]

4.4 Experimental details

Fine-tuning We conducted all of our experiments using Galactica 1.3b model (Taylor et al., 2022), which is a model already pretrained on a large-scale scientific corpus and designed to perform scientific tasks, including scientific QA, mathematical reasoning, document generation, etc. We use Transformer Reinforcement Learning (TRL) library by HuggingFace[‖] to carry out our SFT and DPO trainings with a parameter-efficient technique LoRA with rank=8 and α=16 as suggested by (Hu et al., 2021). We used a cosine learning rate scheduler with a peak learning rate of 2e-6 for SFT and 5e-7 for DPO, as larger learning rate values resulted in NaN loss values during training. We trained the DPO models with a batch size of 4 using β=0.1 following (Tunstall et al., 2023).

Retrieval For chunking our documents, we used a Recursive chunking by Langchain with chunk size of 512 characters considering the context length of our model (2048 tokens). We then used a Context and Question Encoder pairs by Facebook[**] to convert the context and questions into an embedding space. For retrieval, we concatenated each question (Q$_i$) with each of its corresponding options (O$_{ij}$) where $j \in \{1, 2, 3, 4\}$ as [Q$_i$+O$_{i1}$], . . . , [Q$_i$+O$_{i4}$], and retrieved the top 3 documents for each pair. We selected 3 documents out of 12 that had the highest similarity score, where a cosine similarity was used. For efficient similarity score computation, we normalized encoded contexts using their L2 norms and utilized the FAISS library.

[§] Tested in a zero-shot (with no documents in prompt)

[¶] Tested both with documents (+RAG) and without them.

[‖] https://huggingface.co/docs/trl/en/index

[**] facebook/dpr-ctx_encoder-single-nq-base

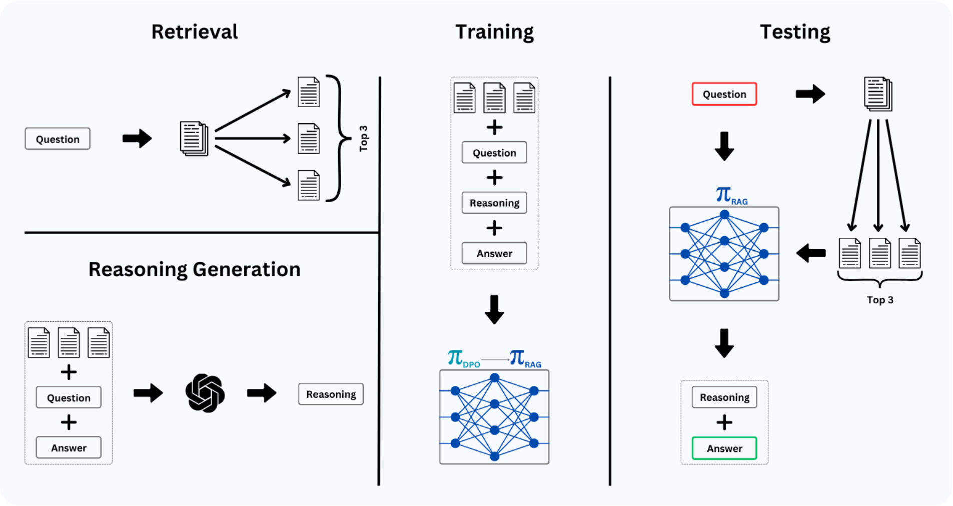
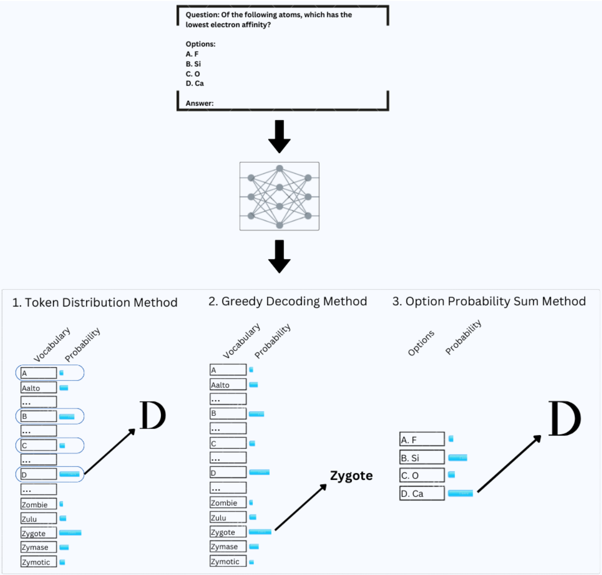


Figure 3: High level workflow of RAG-tuning phase.



Question: Of the following atoms, which has the lowest electron affinity?

Options:
A. F
B. Si
C. O
D. Ca

Answer:

Figure 4: 3 evaluation methods.

4.5 Results

Using the above datasets and baselines, we evaluate our model and present the results in Table 1.

Model	Method	ARC	MMLU	SciQ	EPFL
GalactiTA$_{DPO}$	1	29.24%	29.03%	41.31%	27.32%
	2	20.58%	23.87%	3.51%	18.30%
	3	21.20%	24.52%	37.22%	27.57%
GalactiTA$_{MCQA}$	1	**35.06%**	31.61%	**60.50%**	35.59%
	2	**35.06%**	31.61%	**60.50%**	35.59%
	3	30.87%	29.03%	56.82%	28.07%
GalactiTA$_{MCQA}$ (+ RAG)	1	33.00%	**32.90%**	53.22%	34.59%
	2	33.00%	**32.90%**	53.22%	34.59%
	3	30.87%	28.39%	48.57%	23.81%
GalactiTA$_{RAG}$	1	31.01%	29.68%	47.35%	30.83%
	2	30.94%	29.68%	47.35%	27.57%
	3	25.27%	27.74%	31.84%	31.58%
GalactiTA$_{RAG}$ (+ RAG)	1	29.67%	30.97%	45.47%	31.58%
	2	29.67%	30.32%	45.47%	29.82%
	3	24.34%	25.16%	34.45%	31.83%
TinyLlama	1	24.34%	25.16%	24.00%	36.84%
	2	10.79%	10.97%	17.31%	12.28%
	3	25.48%	28.39%	28.08%	36.59%
TinyLlama$_{MCQA}$	1	26.05%	24.52%	25.63%	27.07%
	2	26.05%	24.52%	25.63%	27.07%
	3	25.69%	27.74%	27.92%	26.57%
TinyLlama (+ RAG)	1	25.20%	22.58%	25.80%	**38.60%**
	2	16.68%	10.97%	20.90%	15.54%
	3	24.34%	27.10%	26.94%	38.35%

Table 1: Accuracy Assessment of Different Models via Three Prediction Methods on a Range of Test Datasets

After MCQA-tuning, we find that the score for method 1 matches that of method 2, showcasing that the model has learned to predict a single letter effectively. When incorporating RAG for testing, we see that, for the EPFL dataset, the GalactiTA$_{RAG}$ model attains higher scores with RAG compared to without and an 11.52% increase is observed between GalactiTA$_{RAG}$(+RAG) and GalactiTA$_{DPO}$ for method 2. This can be attributed to the fact that the retrieval documents were specifically curated to align with EPFL questions. In contrast, for other datasets, the additional context sometimes confuses the model. Indeed, the MMLU dataset also shows a benefit from RAG, however, it's worth noting that MMLU questions cover topics similar to EPFL. Interestingly, we observe that even for the EPFL dataset, TinyLlama is benefiting more from RAG than GalactiTA$_{RAG}$ (across all methods). This is likely because the Tiny-Llama model is instruction fine-tuned to follow directives effectively. It thus uses the provided context from RAG only if it's relevant, otherwise, it ignores it, as we direct it with our added instruction.

5 Analysis

5.1 SFT and DPO trainings

The evaluation scores given in this section correspond to the accuracy of selecting the chosen response[††].

Number of epochs When we trained the SFT model with 1 epoch, we observed that the evaluation loss was still decreasing even towards the end of the training. So we tried training both SFT and DPO with up to 3 epochs. The strongest model was obtained with 1 epoch of SFT followed by 1 epoch of DPO. We also observe that if the SFT model is trained for more than one epoch, the DPO step actually induces a performance regression, as show in Figure 5

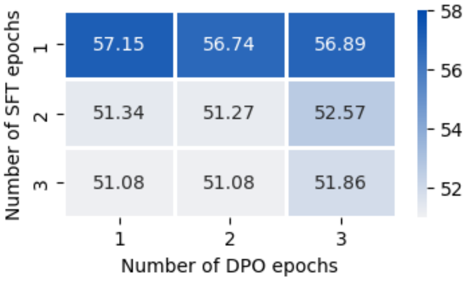


Figure 5: Comparison of Galactica-1.3B models fine-tuned first with SFT followed by DPO for varying number of epochs. The evaluation is done on the EPFL test set.

HUMAN dataset We tried DPO training with only AI dataset and the combination of AI and HUMAN

[††] Defined as the number of times the chosen response is chosen over the rejected response, across the total number of questions in the test set

datasets and found that when trained only on AI, the resulting model performs better on EPFL (AI) test set, but significantly worse on SHP (HUMAN) test set as shown in Table 2. We assume that it is because a larger portion of HUMAN dataset contains discussion-based questions, rather than concrete scientific questions, as they are collected from online platforms such as Reddit. So including those questions in the train set might have worsened the performance of the model on actual scientific questions; See dataset samples in Appendix A.

Dataset	SHP	EPFL
AI	42.83%	**57.15%**
AI + HUMAN	**62.99%**	56.60%

Table 2: Cross evaluation: Galactica-1.3b DPO model on AI and HUMAN datasets.

5.2 RAG-tuning

In the experiments described below, the evaluation score refers to Method 1 (Token Distribution Method). The model flagged as (ours) refers to our GalactiTA$_{RAG}$ model.

Effect of Retrieval We experimented with other ways of retrieving top-k documents. Table 3 compares 5 such methods:

- *Larger Chunks* - a chunk size of 2048 is used. It has the lowest score almost across all datasets, which implies that additional documents in the prompt confuse the model.
- *No stop words* - stop words from the chunks are removed while encoding. By doing so, we tested if the presence of only relevant words in the chunks would improve the embedding quality, hence the retrieval. However it has a lower score than encoding the complete sentences, as, we think, the encoder is trained on the complete natural sentences, instead of a collection of words put together.
- *ST encoder* - *all-MiniLM-L6-v2*[‡‡] model used for encoding both the chunks and the question. Generally, it has a lower retrieval quality, as it is not necessarily trained for question answering.
- *Euclidean distance* - Euclidean distance was used as a similarity score.

Methods	ARC	MMLU	SciQ	EPFL
RAG (ours)	**29.67%**	30.97%	**45.47%**	**31.58%**
Larger chunks	28.60%	28.39%	39.10%	25.06%
No stop words	29.17%	28.39%	42.04%	28.32%
ST encoder	29.38%	**32.26%**	40.08%	26.32%
Euclidean distance	28.53%	29.68%	42.69%	31.08%

Table 3: Comparison of 5 retrieval methods used during inference for the RAG model. RAG (ours) refers to the base retrieval method we are using.

Optimal Top K In Table 4, we experimented with different top-k values for retrieved documents and found that 3 or 4 generally yield better performance. This is likely because higher values help with relevant documents, while lower values are better when documents are irrelevant.

K	ARC	MMLU	SciQ	EPFL
1	28.53%	28.39%	42.37%	32.33%
2	28.89%	29.03%	42.20%	31.83%
3 (ours)	**29.67%**	30.97%	**45.47%**	31.58%
4	29.24%	**31.61%**	43.02%	**34.09%**
5	27.54%	28.39%	43.60%	32.58%

Table 4: Comparison of 5 values for top-k used during retrieval.

Effect of Generator Following (Zhang et al., 2024), we selected several chunks from our document collection, and we used ChatGPT to generate 5 questions (Q) for each chunk and the corresponding CoT answers (A), and created a training set of 8k samples in the format of D^* + D_1 + D_2 + Q -> A, where D^* is the relevant context and D_i are distractor contexts[§§]. We then used this data to train GalactiTA$_{DPO}$ model, which is given as RAFT in Table 5. We also further fine tune our RAFT model with MCQA dataset (RAFT+MCQA). Additionally RAG-CoT is the DPO model trained with the RAG data, but without the CoT responses. As we can see, almost across all datasets, RAFT+MCQA outperforms the RAFT model by a large margin, as the RAFT model by itself is not trained for MCQA. The exception is the EPFL dataset, which appears to benefit from a model trained with the relevant context in the prompt (RAFT). This suggests the retriever works more effectively for EPFL questions. Interestingly, the presence of CoT response in the train set is decreasing the performance, probably due to our decoding method that prevents the

[‡‡] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[§§] The main difference between this data and our RAG data is that, here we are sure that one of the contexts is relevant to the question, but the questions are not of MCQ type.

model from generating a CoT response.

Methods	ARC	MMLU	SciQ	EPFL
RAG (ours)	29.67%	**30.97%**	45.47%	31.58%
RAG-CoT	**30.80%**	25.81%	55.27%	31.83%
RAFT	23.35%	19.35%	25.78%	**32.58%**
RAFT+MCQA	28.89%	25.81%	**55.51%**	30.83%

Table 5: Comparison of 4 methods of RAG generator fine-tuning.

No need for long generations Because our RAG tuning dataset is in the format of: `instruction + context + question + options + reasoning + answer`, we tried with a different decoding method. We fed the model the `instruction + context + question + options` and let it generate 250 tokens for reasoning and answer. We then look for the "Answer:" and use the subsequent token (expected to be a letter from A, B, C, D) as the prediction. From Table 6, we see that this method greatly benefits the SciQ dataset. This is likely due to the nature of the SciQ dataset, where the original data includes a "support" (reasoning) for the answer, which means the questions, by nature, could be answered more accurately with a reasoning step.

Output	ARC	MMLU	SciQ	EPFL
Answer (ours)	29.67%	**30.97%**	45.47%	**31.58%**
Reason + Answer	**33.36%**	24.52%	**62.78%**	29.07%

Table 6: Evaluation of our pipeline on Llama-3 8B model. Llama-3 8b trained with SFT, DPO and either MCQA or RAG

Can larger models help? Finally, to test the effectiveness of our RAG tuning phase we also applied our pipeline on Llama-3 8B base model. We follow the above steps and apply SFT, followed by DPO training. We then apply either the MCQA dataset (Llama$_{MCQA}$) or the RAG dataset (Llama$_{RAG}$) and test them in and out of the RAG setting. From Table 7, we see that when we test the Llama$_{RAG}$ model with RAG, it has significantly a higher performance than testing it without RAG, which confirms that the model has, indeed, learned to respond with the documents in the prompt.

Model	ARC	MMLU	SciQ	EPFL
Llama$_{MCQA}$	75.66%	50.32%	93.14%	48.62%
Llama$_{MCQA}$+RAG	73.39%	50.97%	90.94%	42.86%
Llama$_{RAG}$	33.07%	34.19%	42.37%	32.33%
Llama$_{RAG}$+RAG	62.38%	45.81%	86.20%	40.85%

Table 7: Evaluation of Llama-3 8B on SFT, DPO, followed by either MCQA or RAG-tuning.

6 Ethical considerations

Our model is currently biased towards English, but it can be adapted it to other languages using API-based translation. This approach translates queries before processing and translates responses back. This avoids the high cost of re-fine-tuning for each language and is thus more efficient. It also mitigates data limitations especially for low-resource languages with limited data.

To accommodate DHH (Deaf and Hard of Hearing) students, we can adapt our model to interact in signed language. We can fine-tune it on a subset of the ASL Wikipedia dataset with 254 STEM-related articles, using the remainder for retrieval during inference. However, the scarcity of STEM resources in ASL and the lack of standardized signs for technical terms may limit performance.

Despite potential drawbacks, our model offers 24/7 assistance to students. While this might threaten some TAs' jobs, human interaction remains crucial for students who prefer a social learning environment and thus our model only serves as a complementary resource, helping those who might not otherwise get their questions answered.

Unfortunately, students might misuse the model for direct answers, bypassing the problem-solving process that human TAs promote. This also raises cheating concerns, especially for take-home exams. Professors can mitigate this by warning against AI misuse and using tools to detect AI-generated content.

Finally, we caution against using our model without proper safeguards due to its potential to hallucinate or provide incorrect answers underscoring once again the unlikelihood of human TAs being (entirely) replaced.

7 Conclusion

We demonstrated a training strategy to align an LLM for scientific MCQA through SFT, DPO and RAG-tuning phases. We highlight several crucial design decisions, such as the choice of training data, impact of retrieval and generator parts of RAG, and several decoding strategies for extracting an answer in MCQA setting. To further enhance the ability of the model in RAG settings, some techniques can be tried as a future work: more curated documents collection, reranking strategy during retrieval, and fine-tuning retriever and generator end-to-end.

8 Team contribution

MILESTONE 2

Zied Masmoudi:

- contribution to SFT/DPO data collection
- implementation of SFT training scripts
- training/testing of SFT models
- contribution to report writing

Jakhongir Saydaliev:

- contribution to SFT/DPO data collection
- implementation of SFT/DPO training scripts
- implementation of DPO reward evaluation
- training/testing of DPO models
- ablation studies with DPO models
- contribution to report writing

Mohamed Amine Ben Ahmed:

- contribution to data collection
- documents collection for RAG
- initial implementation of RAG
- contribution to report writing

MILESTONE 3

Zied Masmoudi:

- RAG tuning data collection
- implementation of the first 3 evaluation methods
- training/testing of RAG models
- contribution to report writing (Abstract, Experiments, Ethical considerations)

Jakhongir Saydaliev:

- RAFT data collection
- Ablation studies
- implementation of long generation method
- training/testing of RAG models
- contribution to overall report (focused on Intro, Approach, Analysis, Conclusion)

Mohamed Amine Ben Ahmed:

- preprocessing the documents collection
- implementation of RAG
- testing of RAG (inference)
- contribution to report writing

References

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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.

Hugging-Face. 2023. Open llm leaderboard: Mmlu. `https://huggingface.co/blog/open-llm-leaderboard-mmlu`. Accessed: 2024-06-13.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag.

A Appendix: Dataset

prompt	Which event is the best example of competition between species in a pond environment?\nA. dragonflies landing on lily pads\nB. frogs and toads eating flies\nC. lizards and snakes lying in the sun\nD. hawks eating mice
response	Frogs and toads eating flies is the best example of competition between species in a pond environment. In this scenario, both frogs and toads are competing for the same food source - flies. By consuming the same prey, they are directly competing with each other for resources. This competition can drive natural selection and influence the population dynamics of both species in the pond. In contrast, the other options involve interactions that are not necessarily competitive but rather part of the natural food chain or ecosystem dynamics. So the chosen answer is: B. frogs and toads eating flies

Table 8: A sample of SFT dataset

prompt	Why is skipping masters and going directly for a PhD so common in physics?
chosen	Outside the US progression is 2 yrs Masters + 4 yrs PhD. In the US the progression is 5-7 yrs PhD, Masters don't typically exist except as "consolation prizes" for leaving a PhD early. "Skipping Masters", as you can tell from comments is not really a concept and definitely not a "common" one either in Physics or out of Physics.
rejected	I'm pretty sure it's impossible in Europe.

Table 9: A sample of DPO HUMAN dataset

prompt	Which is the smallest asymptotically? • A: $O(1)$ • B: $O(n)$ • C: $O(n^2)$ • D: $O(\log n)$
chosen	To determine which option is the smallest asymptotically, we need to understand the concept of Big O notation. Big O notation is used to describe the upper bound of an algorithm's time complexity in terms of how it grows relative to the input size. In other words, it describes the worst-case scenario of how the algorithm's time complexity grows as the input size approaches infinity. In Big O notation: • $O(1)$ represents constant time complexity, meaning the algorithm's time complexity does not depend on the input size. • $O(n)$ represents linear time complexity, meaning the algorithm's time complexity grows linearly with the input size. • $O(n^2)$ represents quadratic time complexity, meaning the algorithm's time complexity grows quadratically with the input size. • $O(\log n)$ represents logarithmic time complexity, meaning the algorithm's time complexity grows logarithmically with the input size. Among the given options, $O(1)$ represents the smallest asymptotically, as it indicates constant time complexity that does not depend on the input size. In contrast, $O(n)$, $O(n^2)$, and $O(\log n)$ all grow relative to the input size, with $O(1)$ being the smallest in terms of growth rate. So the chosen answer is: **A.** $O(1)$
rejected	In Big O notation, the complexity classes are ordered from smallest to largest growth rate as follows: $O(1) < O(\log n) < O(n) < O(n^2)$. Therefore, among the options given, $O(1)$ has the smallest growth rate asymptotically. This means that as the input size increases, the time complexity of an algorithm with $O(1)$ will remain constant, making it the smallest in terms of growth rate. So the chosen answer is: **A.** $O(1)$

Table 10: A sample of DPO AI dataset

question	A retrieval model attempts to capture? • A. the interface by which a user is accessing information • B. the importance a user gives to a piece of information for a query • C. the formal correctness of a query formulation by user • D. the structure by which a document is organised
answer	B

Table 11: A sample of MCQ dataset

instruction	Answer the question based on the provided contexts. If the contexts do not provide sufficient information to answer the question, rely on your general knowledge to respond.
contexts	["Retrieving these documents is the second step in the process. This task is known as information retrieval (IR), and IR can be performed...", "Information extraction is the process of acquiring knowledge by skimming a text and Information extraction looking for occurrences...", "The theory suggests that pattern separation is primarily accomplished in the memory encoding process, which starts with..."]
question	A retrieval model attempts to capture? • A. the interface by which a user is accessing information • B. the importance a user gives to a piece of information for a query • C. the formal correctness of a query formulation by user • D. the structure by which a document is organised
reason	The context mentions that in information retrieval (IR), a vector space model is used where word frequencies are represented by vectors. This model focuses on capturing the importance of words in a document for a given query. Therefore, the retrieval model attempts to capture the importance a user gives to a piece of information for a query.
answer	B

Table 12: A sample of RAG dataset

B Appendix: Retrieval

question	Fundamentally, why clustering is considered an unsupervised machine learning technique?
doc 1	Unsupervised machine-learning algorithms aim at findings patterns or internal structures within datasets that contain input data points without labels or targets. K-means clustering, the mixture of Gaussians, and so on are methods of unsupervised learning. Even data-reduction techniques like Principal Component Analysis (PCA), Singular Value Decomposition (SVD), auto-encoders, and so forth are unsupervised learning methods.
doc 2	Clustering is the most well known unsupervised method. Here, we are concerned with making a measurement of similarity between instances in an unlabeled dataset. We often use geometric models to determine the distance between instances, based on their feature values. We can use an arbitrary measurement of closeness to determine what cluster each instance belongs to.
doc 3	Brown clustering (Brown et al. 1992) - Exchange clustering - Desparsification and great example of unsupervised pretraining - So clustering models learn for each cluster/topic a distribution over words of how likely that word is in each cluster - Latent Semantic Analysis (LSA/LSI), Random projections.

Table 13: Top-3 retrieval for an EPFL question sample

question	Which of the following is true regarding the random forest classification algorithm?
doc 1	...algorithm without predictions (blue) when the error is small (but worse when the error is large). A sample illustration of these conclusions is provided in Figure 1 (d), corresponding to the equilibrium with r = 75 and wk = 70. Each point is the average of 1000 samples, with the blue curve corresponding to = 1 and the orange curve to = 0.2.
doc 2	Alternatively, factors can be compressed by representing them using algebraic decision diagrams instead of tables (Gogate and Domingos, 2011). Exact methods based on recursive enumeration (see Figure 13.11) combined with caching include the recursive conditioning algorithm (Darwiche, 2001), the value elimination algorithm (Bacchus et al., 2003), and AND–OR search (Dechter and Mateescu, 2007).
doc 3	training. A phase in which an artificial neural network has its weights adjusted by using backpropagation with known-correct outputs for some given inputs (chapter 7). tree. A graph that has only one path between any two vertices. A tree is acyclic (chapter 4). unsupervised learning. Any machine-learning technique that does not use foreknowledge to reach its conclusions—in other words, a technique that is not guided but instead runs on its own (chapter 6).

Table 14: Top-3 retrieval for an EPFL question sample