Optimization methods in LLMs for Science Multiple Choice QA

Sam Lai	Yichao Yang	Xiaoyu Zhang	Zexuan Yang
Data Science	Data Science	Data Science	Data Science
New York University	New York University	New York University	New York University
jl12560@nyu.edu	yy5020@nyu.edu	xz4535@nyu.edu	zy3035@nyu.edu

Abstract

This research addresses the challenge of selection bias in Large Language Models (LLMs) when applied to multiple-choice question-answering in scientific contexts. We aim to develop an optimized LLM-based system that minimizes bias and enhances accuracy. Our approach combines Supervised Fine-Tuning (SFT), Prompt Engineering, and Retrieval-Augmented Generation (RAG) techniques to improve model performance on science-specific questions. Initial experiments using various Llama model versions demonstrate significant improvements in accuracy and bias reduction, establishing a foundation for further exploration in optimizing LLMs for specialized domains.

1 Introduction

The increasing reliance on Large Language Models (LLMs) for natural language processing tasks presents a significant opportunity for advancements in automated question-answering systems, particularly in scientific domains. However, these models face challenges such as selection bias, where the positioning of answer choices can skew results, and current methodologies often fail to adequately address this issue, leading to inconsistent and unreliable performance in multiple-choice scenarios. Our project tackles these challenges by Supervised Fine-Tuning (SFT) to refine LLMs on domain-specific data while integrating Direct Preference Optimization (DPO) to adjust the model's output rankings. Additionally, we enhance the model's access to real-time data from external sources like Wikipedia, thereby improving its ability. Using approximately 57,000 multiple-choice questions, we demonstrate our optimized model significantly outperforms existing LLMs in both accuracy and bias reduction. This work contributes to the ongoing development of LLM applications in education and research.//

FINDINGS

Motivate the Problem: Begin by explaining why science multiple-choice QA is a challenging yet important task. Discuss how LLMs, despite their general success, struggle in specialized domains like science due to biases in answer selection and limitations in symbol binding.

2 Related Work

Lewis et al. (2020)[1] developed the Retrieval-Augmented Generation (RAG) framework, which forms the basis of many current approaches to knowledge-intensive NLP tasks.

Recent research by Xue et al. (2024)[2] has made significant contributions to addressing selection bias in LLM for multiple-choice questions. They introduced the Point-wise Intelligent Feedback (PIF) method, which constructs negative samples by randomly combining incorrect option contents with all candidate symbols and employs a point-wise loss to provide feedback on these samples during SFT.

Zheng et al. (2024)[3] demonstrate that LLMs exhibit selection bias, meaning they tend to favor certain option IDs (such as "Option A" or "Option C") regardless of the question content.

Ding et al. (2024)[5] examines various strategies that utilize LLMs for data augmentation, highlighting the ability of LLMs to enhance training datasets by diversifying the examples without collecting additional data.

3 Methodology

We began by evaluating the baseline performance of several large language models (LLMs) to identify the most suitable foundation model for fine-tuning, using accuracy as the primary evaluation metric. The models assessed included LLaMA 3.2 3B Instruct, LLaMA 3.2 3B Pretrained, LLaMA 3.1 8B Pretrained, and Mistral-7B Instruct v0.3. To ensure a fair comparison, each model was tested on the same dataset with identical prompts.

Based on evaluation results, we selected the LLaMA 3.2 3B Pretrained model due to its superior accuracy on the test set. To further optimize the model, we aim to address two key challenges: model selection bias and contextual understanding. To mitigate model selection bias, we propose using the Point-wise Intelligence Feedback Supervised Fine-Tuning method [1]. For enhancing contextual understanding, we will implement a retrieval-augmented generation (RAG) system to provide additional context for each question. Since RAG increases token consumption per sample, we prioritize obtaining an unbiased model before integrating RAG to improve training efficiency.

3.1 Prompt Engineering

Inspired by Robinson's paper [2], We compared two different prompting methods which are Cloze prompt and Multiple Choice Prompt.

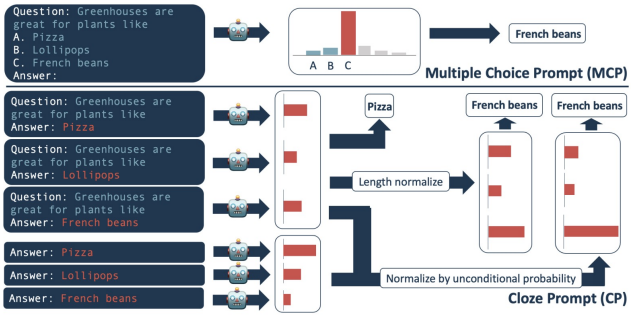

Figure 1: MCP vs CP

Below are the main difference between the two prompting methods:

Aspects	Multiple-Choice Prompting (MCP)	Cloze Prompting (CP)
Likelihood Consideration	No conflation of answer likelihood and language naturalness; focuses solely on the token probabilities for the choices.	Conflates the likelihood of the answer's text as natural language and as a valid answer, which may distort scores, favoring grammatically and stylistically better answers.
Computational Cost	Low: Single forward pass is sufficient. No normalization required.	High: Requires n forward passes for Raw or LN normalization, and $2n$ passes for UN normalization (n options).
Comparison of Options	Direct comparison of all answer options in a single forward pass, enabling the model to contrast between choices.	No explicit comparison; answer probabilities are compared only implicitly through their scores.

Table 1: Comparison of Multiple-Choice Prompting (MCP) and Cloze Prompting (CP).

By applying the multiple choice prompt, we boosted Llama 3.2 3B from 0.32 to 0.56 accuracy and speed up our model inference on 200 multiple choice QAs from 18 mins to 10 mins on an A100 GPU.

3.2 Point-wise Intelligence FeedBack(PIF)

We adopted the approach proposed by Xue et al. [1], who demonstrated that selection bias could arise during the supervised fine-tuning (SFT) stage due to the anchoring effect of label tokens. To address this issue, instead of training the LLM solely on option symbols (e.g., A, B, C), we initialized our model parameters using a dynamic reweighting strategy. This method, referred to as **Reweighted Symbol-Content Binding (RSCB)**, integrates option symbols and their corresponding content into the loss function, yielding the model π_{RSCB}

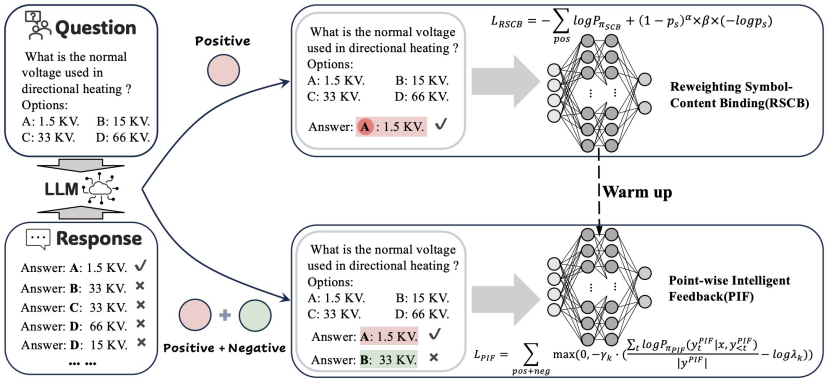

Figure 2: RSCB adjusts the weights of the option symbols and contents in the SFT optimization objective. PIF constructs negative samples by randomly combining the content of incorrect options with all option symbols and designs a point-wise loss to feedback these negative samples into SFT.

The Symbol-Content Binding (SCB) method, where both symbols and answer contents are used as target tokens. The optimization objective becomes:

$$L_{\text{SCB}} = -\sum_{t} \frac{\log P_{\pi_{\text{SCB}}}(y_t^{\text{SCB}} | x, y_{<t}^{\text{SCB}})}{|y^{\text{SCB}}|} \tag{1}$$

$$L_{\text{RSCB}} = L_{\text{SCB}} + (1 - p_s)^{\alpha} \cdot \beta \cdot (-\log p_s) \tag{2}$$

where p_s represents the predicted probability of the correct symbol token, α is a focusing parameter inspired by Focal Loss, and β is the re-assigned weight for the symbol token. By reducing the contribution of well-classified samples, **Focal Loss** emphasizes hard-to-classify samples. In the context of RSCB, if p_s is high, indicating easy prediction, the symbol token's weight is reduced. Conversely, if p_s is low, the model focuses on learning the correct symbol-content association.

Previous studies [3] have explored how to incorporate human feedback with various stages during LLM's training process because it helps LLMs to identify issues with accuracy, fairness, and bias. In the context of MCQs, we possess knowledge of both the positive options' symbols and contents, as well as the negative options. We can also easily acquire negative symbol-content binding examples. We call this process Intelligent Feedback without human preference annotations. Thus, before we obtain the π_{PIF}, we applied negative sampling which we randomly assign negative two samples into our objective

3.3 Retrieval Augemented Generation(RAG)

4 Experiments

4.1 Data

Below is a sample of our test data:

{

```
    "id": "Q8",
    "prompt": "Answer the following question by selecting one of the options. Your response shoul
    "question": "What is the term used in astrophysics to describe light-matter interactions resu
    "options": {
        "A": "Blueshifting",
        "B": "Redshifting",
        "C": "Reddening",
        "D": "Whitening",
        "E": "Yellowing"
    },
    "expected_output_format": "{option}"
}
```


4.2 Evaluation method

Mean Average Precision at 3 (MAP@3) and accuracy. MAP@3 evaluates the model's ability to rank the correct answer among the top three choices.

We will also use answer-moving attack method to calculate the selection bias of our model to ensure we have a robust result. The formula for calculating bias after the answer-moving attack is as follows:

$$\mu_{\text{bias}} = \frac{\sum_{i=1}^{K} |\text{Acc}_i - \text{Acc}_0|}{K}$$

Where:

- μ_{bias} is the calculated bias.
- Acc_i is the accuracy after applying the answer-moving attack on option i.
- Acc_0 is the accuracy on the standard test set without any attack.
- K is the number of answer options in the multiple-choice questions.

4.3 Experimental details

Over the past month, our primary focus has been on model selection for fine-tuning, with particular attention to optimizing the inference process. For the LLaMA 3B models, initial CPU inference time was approximately 700 minutes for 150 questions. This was significantly reduced to 200 mins through the implementation of batch tokenization, which processes all answer options together. Batch tokenization enhances efficiency by enabling parallel processing of options, considerably speeding up inference compared to running each option individually. This approach also prepares the model for even greater efficiency gains during GPU inference.

Following tokenization, we run all options through the model simultaneously to obtain token-level logits. These logits are then converted to log-probabilities using softmax, allowing us to score each option based on the model's confidence. Finally, we rank the options by their total log-probability scores, selecting the highest-ranking option as the predicted answer. For simplicity of our work in model selection, we solely evaluate our model on accuracy for fine-tuning.

We conducted an "answer-moving attack" experiment to assess selection bias of the our best performance model. In this test, we systematically moved all correct answers to each of the options (A, B, C, D, and E) in turn throughout the test set, then evaluated and recorded the model's accuracy after each relocation.

4.4 Results

Report the quantitative results that you have found so far. Use a table or plot to compare results and compare against baselines.

Model	Accuracy
LLaMA 3.2 3B Instruct	14.57
LLaMA 3.2 3B Pretrained	32.21
LLaMA 3.1 8B Pretrained	13.89
Mistral-7B Instruct v0.3	15.23

Table 2: Evaluation Metrics for Different Models

Model	**Accuracy**	**MAP@3**	μ_{bias}
Initial	0.56	0.67	19.88%
RSCB trained	0.925	0.954	14.40%
PIF trained	0.937	0.96	14.12%
RAG	0.963	0.98	13.84%

Table 3: Comparison of Models

5 Future Work

For the next stage, we would like to try different answer attacking methods in our training stage for data augmentation purpose. We also want to try different finetuneing methods such as full finetuneing, Lora and QLora. After that, we would like to enhance our model's knowledge retrieval capabilities with RAG and fine-tuning it with DPO to improve answer selection preferences.

For the model evaluation part, we will also adapt the model to retain and evaluate the top 3 options by log-probability scores to facilitate MAP@3 calculation, providing a more comprehensive performance metric and graph the model selection preference after each training process before and after answer-attacking. This way would better help us to understand and visualize if the model improves using our approaches.

References

[1] Min Xue, Zhiwei Hu, Lei Liu, Kai Liao, Shan Li, Hui Han, Ming Zhao, and Chengqi Yin. Strengthened symbol binding makes large language models reliable multiple-choice selectors. *Association for Computational Linguistics 2024*, 1, 2024.

[2] John Robinson, Curtis M. Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. *International Conference on Learning Representations (ICLR)*, 2023.

[3] Hao Liu, Carlo Sferrazza, and Pieter Abbeel. Chain of hindsight aligns language models with feedback. *International Conference on Learning Representations (ICLR)*, 2024.

[1] Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." In NeurIPS 2020.

[3] C. Zheng, H. Zhou, F. Meng, J. Zhou, and M. Huang, "Large Language Models Are Not Robust Multiple-Choice Selectors," in Proceedings of the International Conference on Learning Representations (ICLR), 2024.

[4] Guo, S., Zhang, B., Liu, T., Liu, T., Khalman, M., Llinares, F., Rame, A., Mesnard, T., Zhao, Y., Piot, B., Ferret, J., Blondel, M. (2024, February 7). Direct Language Model Alignment from Online AI Feedback. arXiv.org. https://arxiv.org/abs/2402.04792

[5] Ding, B., Qin, C., Zhao, R., Luo, T., Li, X., Chen, G., Xia, W., Hu, J., Luu, A. T., Joty, S. (2024, March 5). Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges. arXiv.org. https://arxiv.org/abs/2403.02990

[6] Wang, T., Chen, J., Jia, Q., Wang, S., Fang, R., Wang, H., Gao, Z., Xie, C., Xu, C., Dai, J., Liu, Y., Wu, J., Ding, S., Li, L., Huang, Z., Deng, X., Yu, T., Ma, G., Xiao, H., . . . Zhou, W. (2024, January 30). Weaver: Foundation Models for Creative Writing. arXiv.org. https://arxiv.org/abs/2401.17268