Optimization methods in LLMs for Science Multiple Choice QA

Sam Lai
Data Science
New York University
jl12560@nyu.edu

Yichao Yang
Data Science
New York University
yy5020@nyu.edu

Xiaoyu Zhang
Data Science
New York University
xz4535@nyu.edu

Zexuan Yang
Data Science
New York University
zy3035@nyu.edu

Abstract

Large Language Models (LLMs) are increasingly used for natural language processing tasks, particularly in scientific domains. However, LLMs 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. We aim to develop an optimized LLM-based system that minimizes selection 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. Experiments using various LLM versions demonstrate significant improvements in accuracy and bias reduction, establishing a foundation for further exploration in optimizing LLM for specialized domains like science.

1 Introduction

Despite the remarkable capabilities of LLMs, they still face critical challenges, such as selection bias in multiple-choice tasks, where certain answer options are favored such as (such as "Option A" or "Option C") due to model-specific biases rather than content relevance [1]. Recent research by Xue et al [2] has made significant contributions to addressing selection bias in LLM for multiple-choice questions, using Point-wise Intelligent feedback inspired by RRHF method to help the model identify the positive feedback through data augmentation [3]. For contextual understanding enhancement, the Retrieval-Augmented Generation (RAG) framework [4] provides extensive contextual related information to help the model perform better in knowledge-intensive NLP tasks.

2 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 [2]. Since RAG increases token consumption per sample, we prioritize obtaining an unbiased model before integrating RAG to improve training efficiency.

2.1 Prompt Engineering

Inspired by Robinson's paper [5], We compared two different prompting methods which are Cloze prompt and Multiple Choice Prompt.

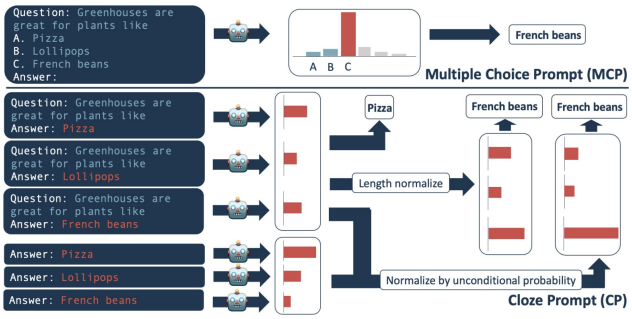

Figure 1: MCP vs CP [5]

Aspects	Multiple-Choice Prompting (MCP)	Cloze Prompting (CP)
Likelihood Consideration	Focuses solely on token probabilities, avoiding conflation of answer likelihood with language naturalness.	Conflates the answer's likelihood as valid text and as a correct answer, potentially favoring more grammatically and stylistically fluent responses.
Computational Cost	Low: Single forward pass is sufficient; no normalization is 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 considered only through their independent 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.

2.2 Point-wise Intelligence FeedBack(PIF)

We adopted the approach proposed by Xue et al. [2], 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}

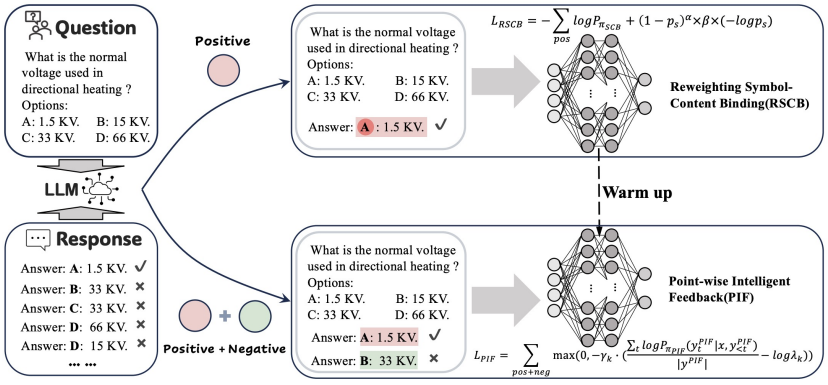

Figure 2: RSCB adjusts the weights of the option symbols and contents in the SFT optimization objective. PIF constructs negative samples by randomly combining the content of incorrect options with all option symbols and designs a point-wise loss to feedback these negative samples into SFT[2]

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 research has demonstrated the effectiveness of incorporating human feedback at various stages of large language model (LLM) training to address issues related to accuracy, fairness, and bias [6]. In the context of multiple-choice questions (MCQs), where both positive and negative option symbols and contents are known, it is possible to generate negative symbol-content pairs without requiring human-annotated preferences. Thus, before we obtain the π_{PIF}, we decided to randomly assign two negative samples to the correct symbol content pair and maximize the probability of positive examples approaching 1 by cross-entropy loss, and minimize the likelihood of negative examples falling below λ[2].

2.3 Retrieval Augemented Generation(RAG)

To enhance the model's contextual understanding, we implemented a RAG system using `txtai` embeddings with a pre-trained Wikipedia index.

The core of our RAG system consists of two key components:

- **Optimized Query Construction:** Combines both the question text and all answer options to create a comprehensive search query, maximizing the relevance of retrieved information for the specific multiple-choice context.
- **Relevance-Scored Retrieval Mechanism:** Returns the k-most relevant text chunks along with their corresponding confidence scores, which are used to weight the importance of each piece of retrieved information.

We empirically set $k = 3$ for context retrieval, finding this to be the optimal balance between information completeness and computational efficiency. The retrieved contexts are integrated with our fine-tuned LLaMA model using the prompt engineering approach described in Section 3.1.

3 Experiments

3.1 Data

Below is a sample of our prompt:

```
"prompt": "Answer the following question based on the {context}
by selecting one of the options.
Your response should be in the format {option}, where {option} is A, B, C, D, or E.",
"context": {Retrieved information}
"question": "What is the term used...options",
"expected_output_format": "{option}"
```

3.2 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.

For selection Bias, We use answer-moving attack technique to calculate the μ_{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.

3.3 Experimental details

We fine-tuned the model using Unsloth API with LoRA ($r = 16$) and Hugging Face's `Trainer` API to balance performance and computational efficiency, reducing VRAM consumption by approximately 30% and speed up the training process by 50% (Support Faster inference as well). The targeting key projection layers such as `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`, with a scaling factor of `lora_alpha=16` to stabilize updates. We applied no dropout (`lora_dropout=0`) and excluded bias terms (`bias="none"`), leveraging optimized settings for reduced memory usage. We used a batch size of `per_device_train_batch_size=2` with `gradient_accumulation_steps=4`, an initial learning rate of 2×10^{-5}, and a linear learning rate scheduler with `warmup_steps=5`. The training process was capped at `max_steps=100` using the `adamw_8bit` optimizer with a weight decay of 0.01. Mixed-precision training was enabled based on BF16 support.

For hyperparameter tuning, we assigned α and β from L_{RSCB} to 2 and 0.1. For λ in the L_{PIF} loss function, we tested it that 0.001 achieves the optimal result.

3.4 Results

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**	$\boldsymbol{\mu}_{\text{bias}}$
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

4 Conclusion

By integrating techniques such as Point-wise Intelligence Feedback (PIF), Reweighted Symbol-Content Binding (RSCB), and Retrieval-Augmented Generation (RAG), we achieved significant improvements in model accuracy and robustness, as evidenced by enhanced MAP@3 scores and reduced selection bias (μ_{bias}).

The results demonstrate that our proposed methods, especially the combination of RSCB and PIF, provide a solid foundation for reducing selection bias in LLMs. Furthermore, the integration of RAG shows the importance of contextual relevance in improving model performance.

References

[1] Cheng Zheng, Hao Zhou, Fanyi Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple-choice selectors. *International Conference on Learning Representations (ICLR)*, 2024.

[2] 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.

[3] Bo Ding, Chuan Qin, Ruijie Zhao, Tian Luo, Xiaobo Li, Guojun Chen, Wei Xia, Jing Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using large language models: Data perspectives, learning paradigms and challenges. *arXiv preprint arXiv:2403.02990*, 2024.

[4] Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Neural Information Processing Systems (NeurIPS)*, 34, 2020.

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

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