Parameter Efficient Fine-Tuning of LLMs towards ASAG

Gugan S Kathiresan, Aditya Shanmugham, Prabhleenkaur Bindra, Maria Anson

CS 6120, Natural Language Processing

April 22, 2024

Abstract

The ability to fine-tune LLMs to different domains has enabled organizations and researchers to adapt their foundation models to their liking. In recent days, Parameter Efficient Fine-Tuning (PEFT) has made this possible on lower computational requirements compared to training LLMs on new domains from scratch. Fine-tuning LLMs have various applications, but a simple way of putting it would be, teaching your LLM to talk in the tone, vocabulary and structure. In this study, we attempt to extend these benefits to the image modality. Visual Question Answering is a popular domain that combines Computer Vision and NLP. We attempt to employ VQA models to identify the components of an image, and employ an LLM adapter to modify the results according to the logical reasoning of the user input question. Through this project, we extend the application to Automated Short Answer Generation(ASAG), to classify and provide an objective feedback to students. We intend to explore various finetuning techniques and datasets to achieve this task.

Keywords— LLMs, ASAG, fine-tuning, PEFT, LoRA

Introduction

Over the recent few years, ASAG has become a benchmark metric for language models, in order to test their performance across standard question answering tasks. ASAG can work as a benchmark for sequence to sequence modeling(providing feedback based on reference answers), it can also serve as benchmark for sequence classification to grade and provide an instant feedback in terms of predictive scoring for students [1]. ASAG also makes sure that we have a consistent and standard grading criteria for all graders. No additional reviews are required by secondary graders, and thus helps in hiring lesser graders, and is also time efficient as grading 600 answers for a particular course is cumbersome and exhausting. The last decade has been exponential and fast-tracked for natural language processing, especially for language and large language models(LLMs). With the rise of transformers, bidirectional encoder representation transformers(BERT), generative pre-trained transformers (GPT), and other modular architectural representations of encoder/decoder models, large language models have gained momentum and is widely used. LLMs are essentially trained on billions of parameters and training a model this big is a cost, compute and time extensive task. Fine tuning an LLM refers to adapting your model to a specific domain; it is specializing your language model for a particular task. Fig 1 describes the different tasks fine-tuning is used for. Parameter Efficient Fine Tuning [2], is a recently introduced concept where in not all parameters and weights in the language model are updated and touched upon. We adapt to the specific domain by freezing most of the weights while reducing the memory usage. Surprisingly, the results of this were comparable to that of full fine tuning. Low Rank Adaptation or LoRA as popularly known, is a technique that involves fixing the pre-trained model weights and introducing trainable rank decomposition matrices into each layer of the Transformer architecture. This method significantly decreases the number of parameters that can be adjusted during downstream tasks.

Fine-tuning is a crucial process because conventional approaches struggle to grasp the full semantic nuances of language. Simple heuristic matching often falls short, especially when explanations lack technical terms. By leveraging the extensive knowledge within data repositories, fine-tuning enables models to better understand language subtleties. Additionally, fine-tuning is essential for adapting models to specific domains, especially when dealing with private data unseen during pre-training. Techniques like PEFT significantly expedite this process, reducing the time required from millions to thousands of instructions for pre-training and fine-tuning, respectively.

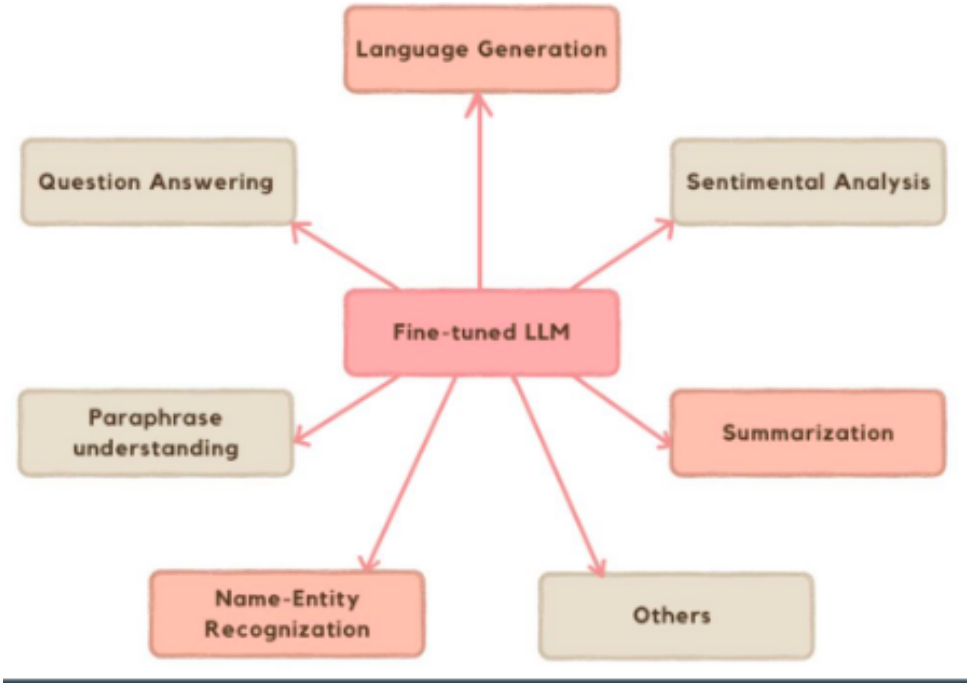


Figure 1: Different tasks finetuning is used for

1 Methods

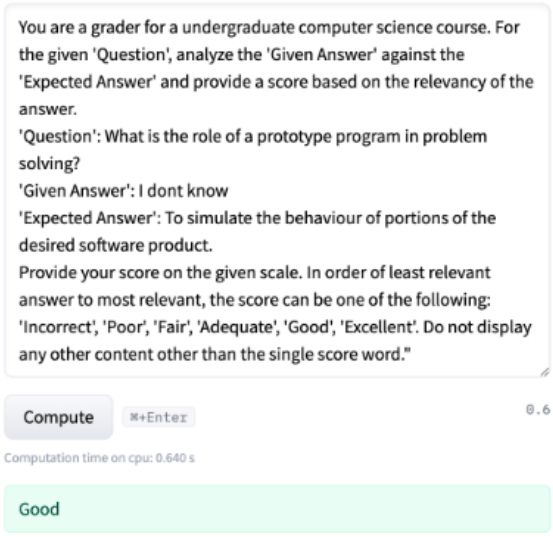
Initially for the project, we attempted prompt engineering and instruction based training for Google Flan-T5 and Gemma models, we got false results for both. Figures 2 below demonstrate the initial performance of these respective models. With the wide base of knowledge that these LLMs cover, we got fairly average performance with quite a few false positives for our dataset. For the project, we compared different LLM models based on their complexity from encoder-only(BERT), encoder-decoder(Flan-T5), and decoder-only(Gemma) architectures. We experimented LoRA instruction based fine tuning on all architectures in order to achieve least training time with the best accuracy.

Our dataset primarily was the Mohler's dataset. The data however was skewed towards positive feedbacks which made us look into more datasets and create a balance between the feedbacks provided. Before combining the two Mohler-like datasets [3, 4], and were subjected to pre-processing that converted numeric scores to string score descriptors. The Mohler dataset is a prominent resource for research in automatic short answer grading (ASAG), particularly within the computer science domain. Compiled by Michael Mohler, it offers a collection of real-world student answer-grade pairs from a computer science course. The dataset encompasses responses from 31 students, likely across 10 assignments with 4-7 questions each, totaling roughly 2273 data points. However, our fully combined dataset contains 5919 unique data points, with 164 unique questions and 4110 unique answers. Figure 3 provides an insight into the distribution of data samples before and after the pre-processing steps.

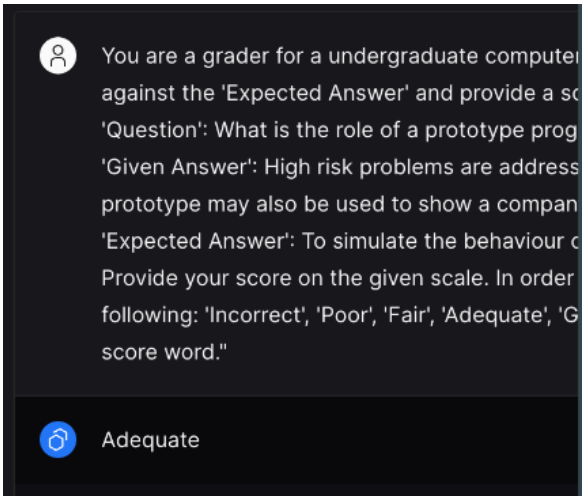
This dataset serves a critical role in the development and evaluation of ASAG models. Researchers leverage it to train algorithms for tasks like pinpointing key concepts, assessing answer quality, and assigning grades based on predefined rubrics. The Mohler dataset also facilitates benchmarking different ASAG models, allowing researchers to compare their effectiveness and identify areas for improvement. Furthermore, it provides valuable insights into the characteristics of well-crafted and poorly constructed student responses within the computer science field. This knowledge can be instrumental in refining teaching methodologies and designing more effective assessment tools.

The dataset was padded with an initial prompt followed by the question and the reference answer as context finally followed by the student's answer, we then ask the model to prompt the objective feedback given the classes a particular answer can be graded as.

In LLMs, understanding the underlying architecture is crucial for selecting the most effective tool for a given task. Our work investigated the performance of three distinct LLM architectures: encoder-only, encoder-decoder, and decoder-only models. Encoder-only models, exemplified by BERT, excel at tasks requiring deep contextual understanding of the input text, such as sentiment analysis or question answering. Their efficiency makes them a valuable starting point for many NLP applications. However, they lack the ability to directly generate text. Conversely, encoder-decoder models, like Flan-T5, possess the power to both comprehend the input and generate an output sequence. This versatility makes them strong contenders for tasks like machine translation or text summarization. However, their complexity comes at the cost of increased computational demands. Finally,



(a) Google Flan - T5 model producing false positive result given context and a dummy student's answer



(b) Gemma 2B outputting false negative result given the context and student answer as prompt with instructions on expected output

Figure 2: Results for ASAG classification before finetuning on different LLM models

decoder-only models, such as Gemma, focus solely on text generation. They can be efficient and demonstrate creative potential in tasks like generating different text formats. However, their reliance solely on a decoder network can limit their ability to grasp the full context of the input, potentially leading to nonsensical or repetitive outputs. By exploring all three architectures, this research aimed to gain a comprehensive understanding of their strengths and weaknesses, ultimately informing the selection of the most suitable LLM for the project's specific goals. Our task in hand was mostly sequence classification and thus we expanded our work majorly towards encoder-decoder and decoder only architectures.

Post our initial analysis and model training, we expanded our knowledge base from text-based dataset to muti-modal datasets, which include handwritten texts, and pdfs to extract student answers from these sources as they are amongst the most common ways students submit their solutions.

2 Experimental Setup

As a preliminary analysis, table 1 describes our results in terms of F1 score for different models.

Our research explored various large language models (LLMs) for a specific task. We investigated models with increasing complexity, ranging from encoder-only (simplest) to encoder-decoder architectures. Examples included Llama v2, Mistral, Mixtral, FLAN UL2, and Gemma (all trained with half precision except Gemma). The total training time exceeded 35 hours at a cost ratio of approximately $50-$60.

Our findings suggest several factors influencing Gemma's performance: RoPE embeddings, GeGelu activation function, and the use of Lora (Low Rank Adaptation) on all linear layers instead of just attention components.

We also explored the relationship between model rank, alpha (hyperparameter), model size, and dataset size. The results suggest a sweet spot for rank between 8 and 16, with alpha typically at 4 times the rank. This deviates slightly from some research suggesting a 2x alpha. For our experiments we ran each of the models to 20 epochs with rank ranging from 16 to 128.

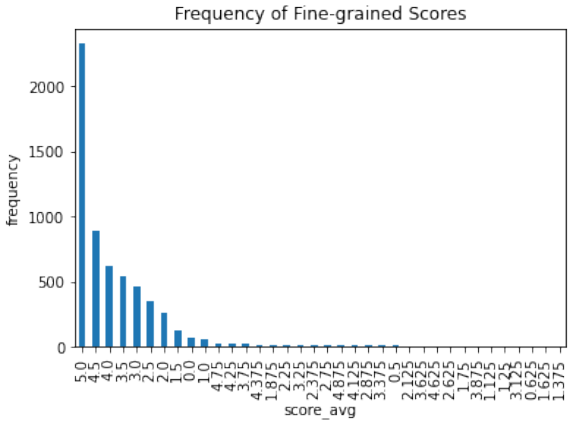
Learning rate also plays a crucial role. While research indicates a 5x increase for Lora learning rate, we observed a 3x increase with weight decay yielded the best results for our specific model. The learning rate for full fine tuning was 1e^-4 while that for LoRA fine tuning was set to a little higher due to parameters being reduced.

Interestingly, we found that classification tasks outperformed regression approaches for this problem, suggesting sequence classification might be better suited than causal modeling techniques.

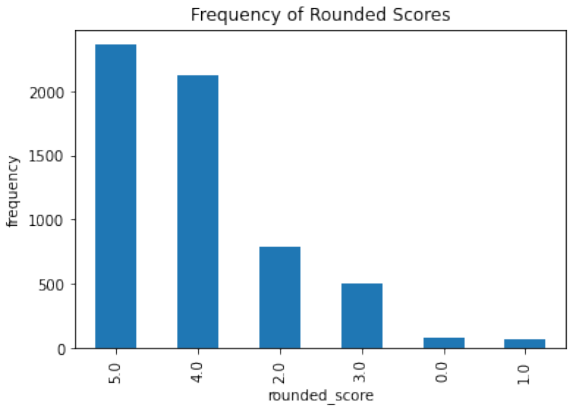
Finally, instruction finetuning with prompt guardrails proved to be an effective approach, even for pre-trained models without further fine-tuning. This highlights the potential of this technique for leveraging pre-trained LLMs.

3 Results

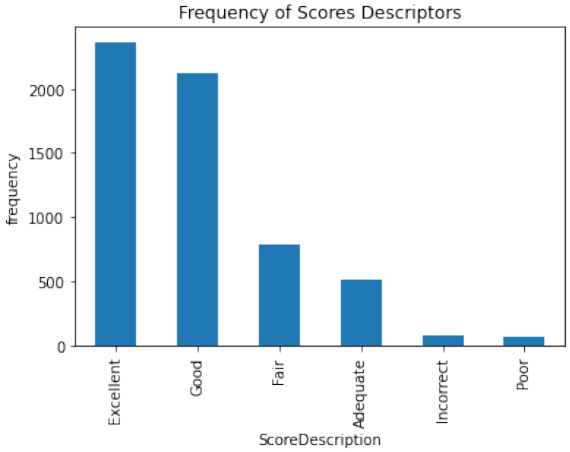
Our investigation focused on identifying the optimal large language model (LLM) for a specific task involving an imbalanced dataset. We employed F1-score, the harmonic mean of precision and recall, to evaluate model



(a) Fine-grained scores



(b) Rounded scores



(c) Score Descriptors

Figure 3: Dataset distribution

performance due to the imbalanced nature of the data.

Initial experiments using a 250M parameter FLAN-T5 model achieved an F1-score of 77% when fine-tuning all parameters. While this demonstrated promising results, we sought a more efficient approach.

LoRA (Low Rank Adaptation) fine-tuning emerged as a viable alternative. While achieving performance close to the full fine-tuning approach, LoRA offered significant training time reductions (approximately one-third) and the ability to train on a smaller subset of parameters (ranging from 1% to 10% depending on the chosen rank). This translates to substantial cost savings.

We explored larger models like Gemma (2B parameters), but the goal was to strike a balance between model size and training cost. While larger models often embody greater knowledge, the cost associated with training them, coupled with potential performance degradation compared to smaller models, rendered them less practical in this scenario.

Gemma (2B parameters) presented an ideal compromise. Despite incurring a 30x training cost compared to FLAN-T5, it offered a significant reduction in training time and comparable performance. This balance between cost and performance makes Gemma a compelling choice for this specific task and dataset.

Weighted Metrics/Model	Encoder Only	Encoder-Decoder			Decoder Only
	BERT	**Flan-T5 Base**			**Gemma**
	SVM Classifier	No Finetuning	Full Finetuning	LoRA finetuning	LoRA finetuning
# of Params	110 M	248 M			2 B
F1 Score	68.15%	0.03%	77.52%	71.74%	**90.94%**
Precision	68.06%	0.01%	78.29%	72.15%	92.34%
Recall	69.59%	1.29%	77.81%	72.57%	90.87%

Table 1: Model comparison based on number of parameters and F1 score

4 Discussion and Conclusion

In conclusion, our research highlights the importance of considering both performance and training cost when selecting an LLM for imbalanced datasets. LoRA fine-tuning offers a promising avenue for achieving competitive performance with reduced training times and costs. When model size necessitates significant cost increases without commensurate performance gains, smaller models like Gemma can provide an effective solution.

5 Github Repository - Codebase

The code and dataset used in this project are available at the following link to the official GitHub repository: LORA LLM Instruction Finetuning for ASAG

6 Contributions

For the project, Anson worked throught EDA and the BERT model. Aditya explored decoder only models included Gemma and LLama. Gugan and Prabhleen worked through the Flan-T5 models for full and LoRA finetuning. All of us equally contributed through for the presentation and documentation.

References

[1] G. Kortemeyer, "Performance of the pre-trained large language model gpt-4 on automated short answer grading," 2023.

[2] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," 2023.

[3] wuhan 1222, "Asag/asag method/dataset/northtexasdataset/expand.txt." GitHub, 2024. https://github.com/wuhan-1222/ASAG/blob/main/ASAG%20Method/dataset/NorthTexasDataset/expand.txt.

[4] G. Kolappan, "Ganesamanian/computer-assisted-short-answer-grading-with-rubrics-using-active-learning." GitHub, Aug. 09 2022. https://github.com/Ganesamanian/Computer-Assisted-Short-Answer-Grading-with-Rubrics-using-Active-Learning/tree/master.