PROJECT REPORT

AHMAD HAKIMI BIN ADNAN

Table of content

Project Title

A Comparative Study of Fine Tuning Technique on a Large Language Model.

1. Introduction

Large Language Models (LLMs) like GPT and LLaMA have shown good performance in various natural language processing tasks such as text summarisation and sentiment analysis. However, their massive size often makes them expensive and inefficient to fine-tune on custom or domain-specific datasets. To address this challenge, various fine-tuning techniques have been introduced over the years ranging from full fine-tuning, where all model parameters are updated, to transfer learning, where knowledge from a pre-trained model is reused for a new task. Each technique has its own advantages and limitations, depending on the specific application and resource constraints.

This project focuses on comparing multiple fine-tuning approaches using 3 different baseline models and one dataset. Specifically, three fine-tuning techniques will be evaluated including Low-Rank Adaptation (LoRA), Quantized Low-Rank Adaptation (QLoRA) and Top Layer which two of these techniques fall under the category of Parameter-Efficient Fine-Tuning (PEFT). The Parameter-Efficient Fine-Tuning (PEFT) is a technique that fine-tunes models by updating only a small portion of parameters while keeping the majority of the model unchanged. In addition, Top-Layer Partial Fine-Tuning will be included as one of the least expensive baseline for comparison. All these techniques will be applied to 3 different pre-trained LLMs across one domain specific dataset. The goal is to assess the effectiveness, computational efficiency, and compatibility of each method in adapting LLMs to tasks under varying conditions.

2. Literature Review

Large Language Models (LLMs) have rapidly become important in natural language processing (NLP) due to their ability to generalize across multiple tasks. The movement toward pre-trained transformer-based models began with Vaswani et al.'s (2017) introduction of the Transformer architecture, which laid the foundation for models like BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and later, LLaMA (Touvron et al., 2023). These models are pre-trained on massive corpora and can be fine-tuned to perform specific tasks such as sentiment analysis, question answering, and summarization.

Early fine-tuning strategies involved full fine-tuning, where all the model's parameters are updated during training on a new task (Howard & Ruder, 2018). While effective, full fine-tuning is computationally expensive, especially for large models with billions of parameters. This led to the emergence of transfer learning as a practical alternative, where models are pre-trained once and reused across tasks, minimizing redundant computation (Pan & Yang, 2010).

To further reduce the computational burden, Parameter-Efficient Fine-Tuning (PEFT) methods have been developed. These approaches allow the adaptation of LLMs by modifying only a small subset of parameters while keeping the rest of the model unchanged. One of the most notable PEFT techniques is LoRA (Low-Rank Adaptation), introduced by Hu et al. (2021). LoRA works by injecting trainable low-rank matrices into existing weight layers, significantly reducing the number of parameters needed for fine-tuning while maintaining performance.

Building upon LoRA, QLoRA (Quantized LoRA) was proposed by Dettmers et al. (2023), which further improves memory and storage efficiency by applying quantization to the base model. This enables fine-tuning of large models on consumer hardware with significantly lower memory footprints, without compromising model quality. This approach has proven to be parameter-efficient, making it suitable for multi-task learning and deployment.

Lastly, Top-Layer Fine-Tuning, also known as partial fine-tuning, involves training only the final classification layer of a model while keeping the rest of the model unchanged. Though it is not considered a PEFT technique in the strict sense, it serves as a common baseline for comparison due to its simplicity and low resource demands (Peters et al., 2019). While each of these methods has been studied individually, few works compare them across multiple LLM architectures and varied NLP tasks. This gap presents an opportunity to evaluate their trade-offs in terms of performance, efficiency, and practical usability which this project aims to address.

3. Research Questions

1. How do different fine-tuning techniques affect the performance of different Large Language Models?

2. Which fine-tuning technique offers the best balance between performance and resources when adapting to question-answering tasks?

4. Research Objective

1. To fine-tune a pre-trained Large Language Model on a domain specific dataset.
2. To understand and implement the LoRA, QLoRA and top layer partial fine-tuning mechanism on a pre-trained LLM.
3. To evaluate the performance of the fine-tuned models using task-specific metrics.
4. To assess the computational resources consumed during the fine-tune process.

5. Methodology

a. Model Selection

For this project, a baseline large language model (LLM) with open access, lightweight and strong community support is selected to ensure ease of implementation and reproducibility. Suitable models such as Llama offer a good balance between performance, making them ideal for experimentation in a fine-tuning pipeline. Below are three selected models:

1. LLaMa 3.2 (1B)

LLaMA 3.2 1B is a small, open-source language model released by Meta in September 2024 under a permissive license. With 1 billion parameters, it is optimized for running efficiently on edge devices and low-resource hardware while maintaining strong language understanding capabilities. The model supports a wide variety of natural language tasks, including multilingual question answering, instruction following, and summarization, making it versatile for smaller-scale QA fine-tuning projects.

Despite its compact size, LLaMA 3.2 1B performs well on reasoning and instruction alignment benchmarks, offering a good trade-off between accuracy and resource usage. Its compatibility with parameter-efficient fine-tuning (PEFT) techniques like LoRA and QLoRA allows adaptation to

QA datasets without requiring large-scale computing infrastructure. Its multilingual support enables fine-tuning for question-answering in diverse languages, making it ideal for rapid experimentation and deployment in constrained environments.

2. Llama 3.2 (3B)

LLaMA 3.2 2B is a mid-sized member of Meta's third-generation LLaMA family, also released in September 2024 under a permissive license. Containing 3 billion parameters, it offers a balance between performance and efficiency, delivering stronger reasoning capabilities and broader generalization than the 1B variant. It is designed to handle a range of NLP applications, including multilingual question answering, summarization, and dialogue-based instruction following, making it highly suitable for QA-focused fine-tuning tasks.

The 3B version excels in instruction adherence and knowledge recall, benefiting open-domain QA systems that require both factual accuracy and contextual understanding. Its support for PEFT methods such as LoRA and QLoRA enables cost-effective fine-tuning even on modest hardware setups. Additionally, its multilingual capabilities expand the scope of dataset experimentation, allowing researchers to fine-tune QA systems that work across various languages and domains while maintaining manageable compute demands.

3. Mistral 7B

Mistral 7B is an open-source large language model developed by Mistral AI, released in September 2023. With 7.3 billion parameters and a 32,000-token context window, it is widely recognized for outperforming many larger models like LLaMA 2 13B in tasks such as question answering, summarization, and general reasoning. Mistral 7B is fully open under the Apache 2.0 license, allowing for commercial use and fine-tuning, making it a good choice for projects with limited resources. It is highly efficient, capable of running on a single consumer GPU. Its compatibility with LoRA and QLoRA makes it especially suitable for fine-tuning on domain-specific QA datasets. Mistral 7B has strong instruction-following ability, fast inference speed, and ease of deployment through platforms like Hugging Face. It remains one of the best lightweight models for real-world QA systems.

b. Tools and Frameworks

The development and fine-tuning processes will be conducted in a Jupyter notebook because Jupyter Notebook provides an interactive interface and supports real-time visualization and most importantly integrates with Python-based machine learning libraries. All necessary libraries, including Hugging Face's Transformers and PEFT will be installed within the local environment. Hugging Face's Transformers library is used for model loading. The Parameter-Effiecient Fine Tuning (PEFT) library supports the application of LoRA, allowing selective training of model components. Following is the explanation of each libraries:

1. Hugging Face's Transformer

   Hugging Face's Transformer library is used for loading and working with pre-trained large language models such as LLaMA and Mistral. It provides simple access to model architectures, tokenizers, and fine-tuning workflows, and is widely adopted in both academic and industry settings. This library also supports integration with parameter-efficient fine-tuning methods (PEFT), which is used in this project.

2. Parameter-Efficient Fine-Tuning (PEFT)

   The PEFT library from Hugging Face, is used to manage fine-tuning techniques such as LoRA, QLoRA, and Top Layer Tuning. This library allows for modular integration of tuning strategies by wrapping the model layers selectively, making it easy to switch between methods and compare their performance. It also ensures compatibility with popular pre-trained models available through Hugging Face.

3. PyTorch

   In the fine-tuning code, PyTorch (torch) is used as the core deep learning framework that powers the model operations. It's responsible for handling tensors (the data structures that store numerical values for the model), moving data to the correct device (GPU), and managing precision types like torch.float16 or torch.float32 to optimize speed and memory usage. PyTorch is also used to control model parameters, such as freezing layers by setting **param.requires_grad = False**, and enabling certain features like gradient_checkpointing to save memory during training. During evaluation and text generation, PyTorch ensures that computations run efficiently on your hardware by handling operations like

.to(model.device) and using torch.no_grad() to disable gradient tracking for faster inference.

4. Weight & Biases (wandb)

Lastly, Weights & Biases (wandb) is used as a tracking and logging tool to monitor the training process. It records key metrics such as the number of trainable parameters, training loss, and evaluation scores in real time, allowing me to visualize and compare experiments through the W&B dashboard. W&B also integrates with the Hugging Face Trainer so that during training, it automatically logs metrics like loss curves and evaluation results, making it easier to analyze performance trends and debug the model.

c. Data Preparation

The dataset for fine-tuning consists of input-output instruction pairs, sourced from open datasets. Input-output instruction pairs is when each datapoints consists of an input (a task or questions given to the model) and output (the answer from the model). These pairs are crucial for training the model to understand and follow human commands. That is how Alpaca-style synthetic instructions serve as a quick-start option. To facilitate a fast fine-tuning process, Alpaca-style synthetic instructions which is a publicly available set of AI-generated instruction-response examples are used. This dataset provides a strong baseline for instruction-tuning, making it ideal for building conversational and task-solving models.

For this project, the Chinese Military Entration Dataset (CMNEE) is utilized to conduct the experiments. The dataset, originally in Chinese and stored in a JSON file, was translated into English using a translation model. To facilitate the fine-tuning process, it was then reformatted into an Alpaca-style structure."

d. Fine-Tuning Process

The fine-tuning process begins by loading the selected baseline LLM and then applying three different parameter-efficient fine tuning techniques including LoRA (Low-Rank Adaptation), QLoRA (Quantized Low-Rank Adaptation) and Top Layer Tuning.

Firstly, LoRA is applied using the PEFT library, which modifies the model to include low-rank adapter layers. Only these LoRA layers are trained during the process, while the original model weights remain unchanged. This approach reduces computational resources. After training, the LoRA adapters are saved separately, enabling deployment. Upon completion, the trained LoRA adapters are saved separately from the base model, allowing for modular deployment or reuse without modifying the original model.

Next, top-layer tuning. Top layer tuning is a simple and lightweight fine tuning technique where only the output layers of a pre-trained model are trained, while the remaining hidden layers in the model are unchanged. This technique is chosen as a baseline for its minimal computational requirements and quick implementation. It is particularly useful in low-resource environments or when rapid prototyping is needed. The main idea is to keep the knowledge of the base model untouched while only adjusting the last layer to better fit the specific task. Eventhough, in some studies it shows that the performance gain is limited compared to more expensive methods, it provides a practical and cost-efficient benchmark.

Lastly, QLoRA (Quantized Low-Rank Adaptation) tuning which is an extension of LoRA by adding quantization to the base model and reducing the precision to 8-bit. This makes the model smaller and more efficient to load and fine-tune. QLoRA is chosen because it has the capabilities of fine-tuning a large model on limited hardware. QLoRA works by applying quantization to compress the base model's weight, then using LoRA to inject small trainable low-rank layers for taks-specific learning. This will result in a very low usage of memory without a loss in performance.

e. Evaluation

To evaluate the performance of various fine-tuning techniques that applied to large language models (LLMs), this project applies a two phase evaluation strategy which is firstly assessing the capabilities of the base models before fine-tuning, and next comparing the improvements achieved after applying different fine-tuning methods. This approach ensures a clear understanding of how much value each technique adds, both in terms of performance gains and resource efficiency. The evaluation process will evaluate three different models capabilities before and after fine tuning and also evaluate with one datasets in which at the end, there will be 9 different results.

In the first stage of evaluation, the base large language models (LLMs) are assessed in their zero-shot or pretrained form using standard question answering or text generation tasks. The evaluation employs four complementary metrics F1, BERTScore, BLEU, and ROUGE. The F1 score measures token-level overlap between predicted and ground truth answers by balancing precision (the proportion of predicted tokens that are correct) and recall (the proportion of correct tokens that were predicted). This makes it useful when answers are partially correct, although it cannot detect meaning beyond exact token matches. BERTScore, on the other hand, evaluates similarity at the semantic level by comparing embeddings from a pretrained language model, allowing it to recognize that "The Eiffel Tower is in Paris" and "Paris hosts the

Eiffel Tower" mean the same thing despite different wording. BLEU measures n-gram precision and applies a brevity penalty, making it effective for tasks where exact phrasing is important, but it can penalize valid answers that use synonyms or alternate sentence structures. Finally, ROUGE focuses on recall-oriented n-gram and sequence matching (ROUGE-1, ROUGE-2, ROUGE-L), making it well-suited for evaluating coverage in longer outputs like summaries, though it too operates at a surface level without deep semantic understanding. Together, these metrics provide a balanced evaluation, F1 for partial correctness, BERTScore for meaning, BLEU for exact phrase precision, and ROUGE for coverage of key content.

In the second stage, each model is fine-tuned using three different parameter-efficient fine-tuning methods which are LoRA (Low-Rank Adaptation), QLoRA, and Top-Layer Tuning. After fine-tuning, the same performance metrics are applied to measure improvement. In addition, to assess computational efficiency, several resource-based metrics are included including the number of trainable parameters, training time, inference latency, peak memory usage, and final model size on disk. These metrics are captured using PyTorch utilities, Python's time module, torch.cuda for memory tracking, and os for storage size measurement. Lastly, Tensorboard will be used to visulize all quantitative evaluation through interactive charts and graphs, making the result easy to comprehend and compare. This combination allows fair comparison of each fine-tuning method not only in terms of performance, but also in terms of practicality, cost, and scalability when deployed in resource-constrained environments. For instance, LoRA and QLoRA techniques are expected to show favorable results in terms of lower memory consumption and fewer trainable parameters compared to traditional fine-tuning.

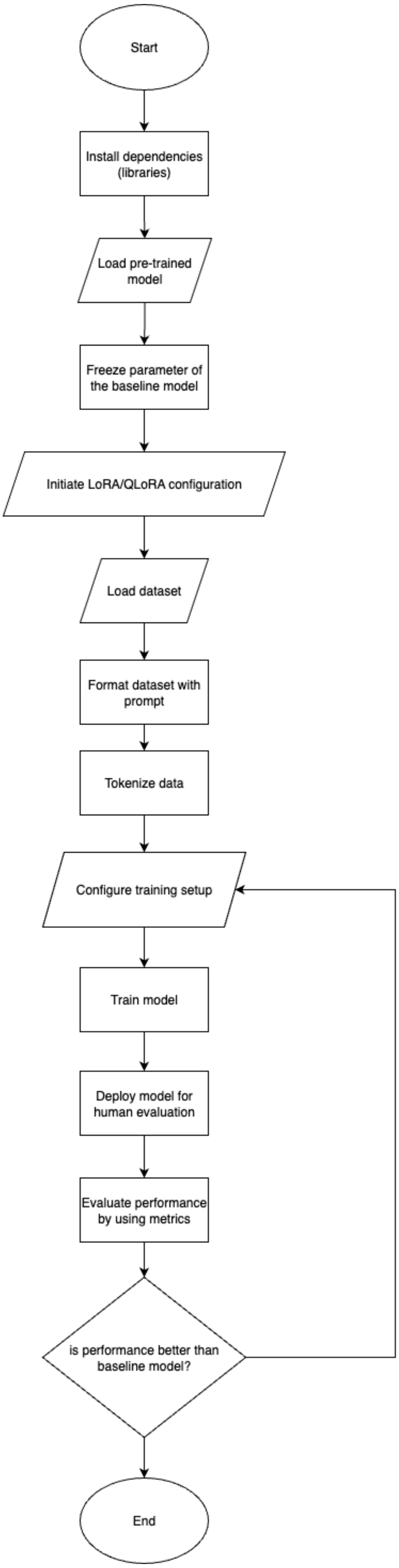
Metrics	Usage
F1, BLEU, ROUGE	Performance Evaluation
BERTScore	Answer and semantic quality
Trainable parameters, power usage, memory usage	Computational efficiency

Table 1: Evaluation Metrics and usage table

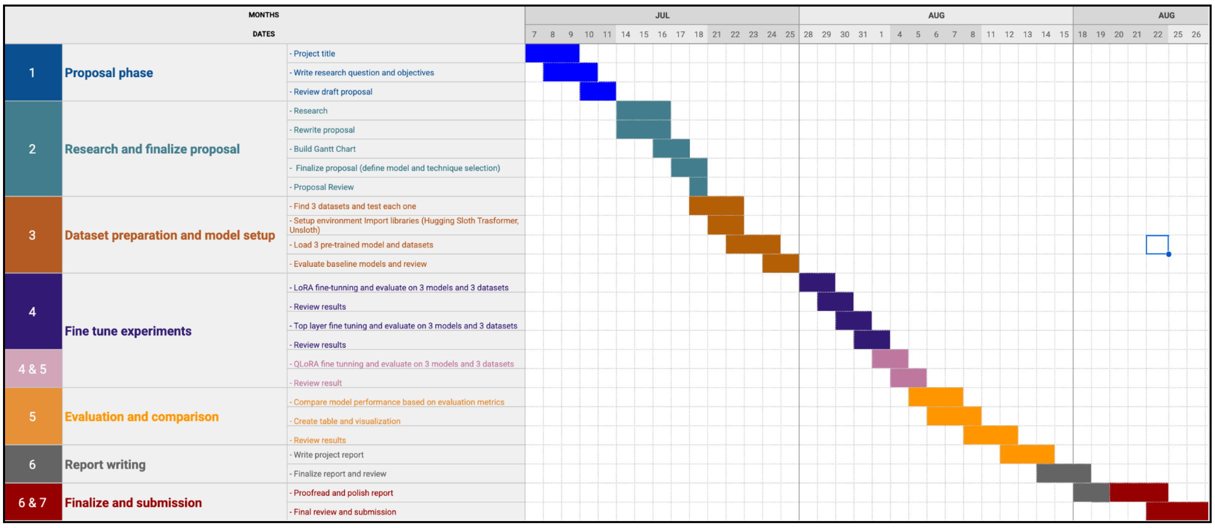
Comparison Table of Evaluation Metrics

Metric	Focus	Measures	Strengths	Limitations
F1	Token-level match.	Harmonic mean of precision & recall.	Good for partial matches, balances accuracy and coverage.	No semantic understanding, exact token match only.
BERTScore	Semantic similarity.	Cosine similarity of embeddings.	Captures meaning even with different wording.	Computationally heavier, depends on the quality of embeddings.
BLEU	Precision-oriented n-gram match.	N-gram precision with brevity penalty.	Well-suited for exact-phrase-sensitive tasks.	Penalizes synonyms and rewording, less recall-focused.
ROUGE	Recall-oriented n-gram & sequence match.	Unigram, bigram, and sequence recall.	Good for content coverage in long outputs.	Still surface-level, ignores deep meaning.

6. Flowchart


Start
Install dependencies (libraries)
Load pre-trained model
Freeze parameter of the baseline model
Initiate LoRA/QLoRA configuration
Load dataset
Format dataset with prompt
Tokenize data
Configure training setup
Train model
Deploy model for human evaluation
Evaluate performance by using metrics
is performance better than baseline model?
End

7. Project Timeline



1	Proposal phase	- Project title
		- Write research question and objectives
		- Review draft proposal
2	Research and finalize proposal	- Research
		- Rewrite proposal
		- Build Gantt Chart
		- Finalize proposal (define model and technique selection)
		- Proposal Review
3	Dataset preparation and model setup	- Find 3 datasets and test each one
		- Setup environment Import libraries (Hugging Sloth Trasformer, Unsloth)
		- Load 3 pre-trained model and datasets
		- Evaluate baseline models and review
4	Fine tune experiments	- LoRA fine-tunning and evaluate on 3 models and 3 datasets
		- Review results
		- Top layer fine tuning and evaluate on 3 models and 3 datasets
		- Review results
4 & 5		- QLoRA fine tunning and evaluate on 3 models and 3 datasets
		- Review result
5	Evaluation and comparison	- Compare model performance based on evaluation metrics
		- Create table and visualization
		- Review results
6	Report writing	- Write project report
		- Finalize report and review
6 & 7	Finalize and submission	- Proofread and polish report
		- Final review and submission

Figure 1: Gantt Chart Internship Project timeline (Version 2)

8. Expected Outcome

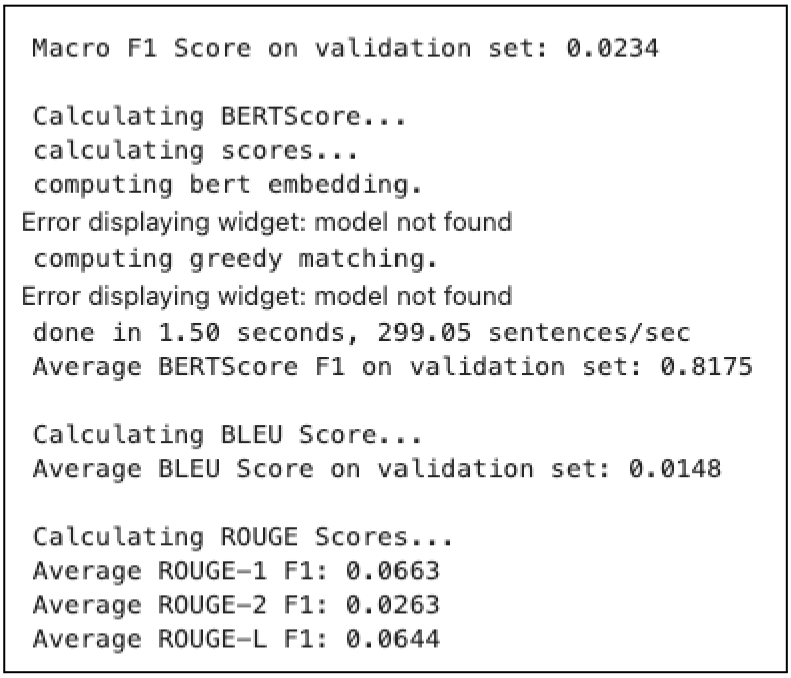
This project aims to produce a fully functional pipeline for fine-tuning a large language model (LLM) using LoRA, QLoRA and Top layer fine tuning techniques. The expected result is that QLoRA produces the smallest and more cost-efficient fine-tuned model that maintains strong performance on the datasets. Additionally, the project will provide insights into the effectiveness of QLoRA, LoRA and top layer fine tuning in low-resource environments, showing its benefits over traditional fine-tuning methods.

9. Result

Llama 3.2 1B

Baseline model

https://wandb.ai/ahmadhakimiadnan-other/baseline-gpu-monitor/runs/8o3ty99m/overview



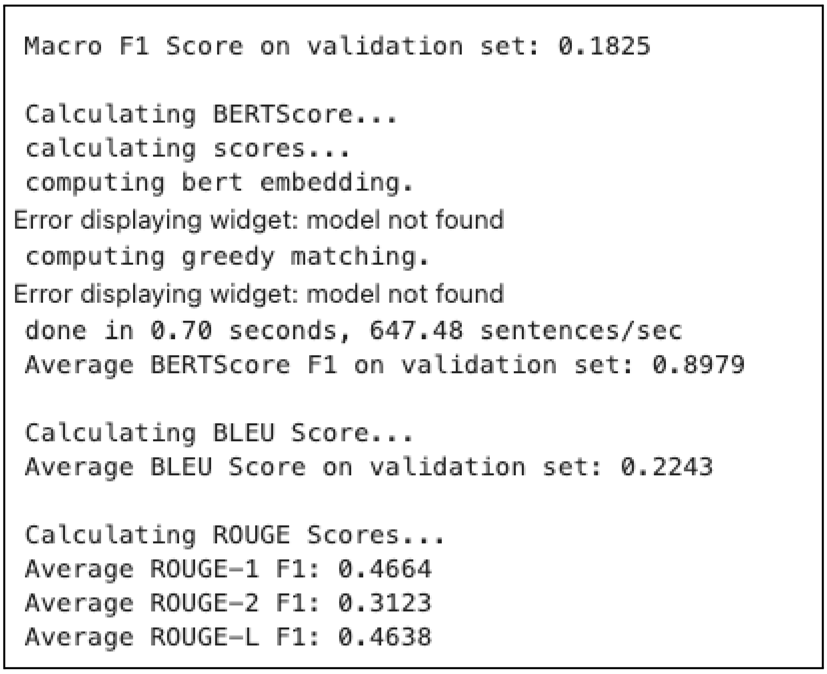
```
 Macro F1 Score on validation set: 0.0234

 Calculating BERTScore...
 calculating scores...
 computing bert embedding.
Error displaying widget: model not found
 computing greedy matching.
Error displaying widget: model not found
 done in 1.50 seconds, 299.05 sentences/sec
 Average BERTScore F1 on validation set: 0.8175

 Calculating BLEU Score...
 Average BLEU Score on validation set: 0.0148

 Calculating ROUGE Scores...
 Average ROUGE-1 F1: 0.0663
 Average ROUGE-2 F1: 0.0263
 Average ROUGE-L F1: 0.0644
```

The evaluation results indicate that the LlaMA 3.2 1B baseline model performed poorly overall on the validation set across most metrics. The Macro F1 Score was 0.0234, suggesting the model struggled with balanced performance across all classes. In terms of semantic similarity, the BERTScore F1 was relatively higher at 0.8175, indicating the generated text had some degree of contextual overlap with the reference outputs. However, the BLEU Score was only 0.0148, reflecting very low n-gram precision. Similarly, the ROUGE metrics showed weak performance, with ROUGE-1 F1 at 0.0663, ROUGE-2 F1 at 0.0263, and ROUGE-L F1 at 0.0644, meaning the model failed to capture much lexical or sequential similarity to the target text. Overall, while the semantic similarity metric (BERTScore) appears acceptable, the low F1, BLEU, and ROUGE scores point to limited accuracy and fluency in the generated outputs.

LoRA

Sources : https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/2hqurf4i/overview

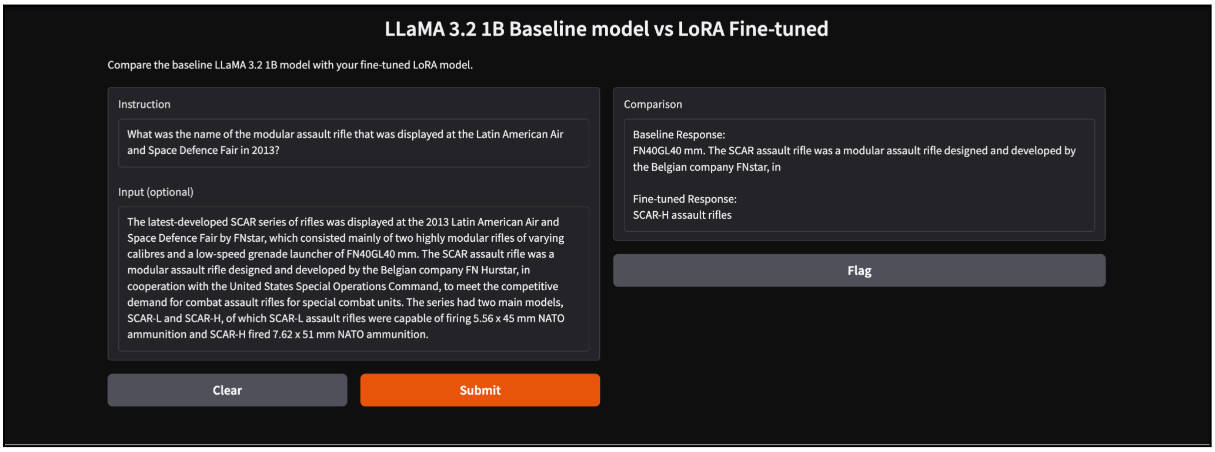
Metrics Evaluation



```
 Macro F1 Score on validation set: 0.1825

 Calculating BERTScore...
 calculating scores...
 computing bert embedding.
Error displaying widget: model not found
 computing greedy matching.
Error displaying widget: model not found
 done in 0.70 seconds, 647.48 sentences/sec
 Average BERTScore F1 on validation set: 0.8979

 Calculating BLEU Score...
 Average BLEU Score on validation set: 0.2243

 Calculating ROUGE Scores...
 Average ROUGE-1 F1: 0.4664
 Average ROUGE-2 F1: 0.3123
 Average ROUGE-L F1: 0.4638
```

After fine-tuning the model with LoRA, there was a significant improvement across all evaluation metrics on the validation set. The Macro F1 Score increased to 0.1825, indicating better overall classification balance compared to the previous result. The BERTScore F1 also rose to 0.8979, showing stronger semantic similarity between the generated outputs and the reference texts. In terms of surface-level text matching, the BLEU Score improved to 0.2243, reflecting higher n-gram precision. Substantial gains were also observed in the ROUGE metrics, with ROUGE-1 F1 at 0.4664, ROUGE-2 F1 at 0.3123, and ROUGE-L F1 at 0.4638, all of which suggest that the model now captures more relevant words, phrases, and sequence structures from the target text. Overall, LoRA fine-tuning enhanced the model’s ability to produce more accurate, fluent, and contextually aligned outputs.

Human Evaluation



LLaMA 3.2 1B Baseline model vs LoRA Fine-tuned

Compare the baseline LLaMA 3.2 1B model with your fine-tuned LoRA model.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

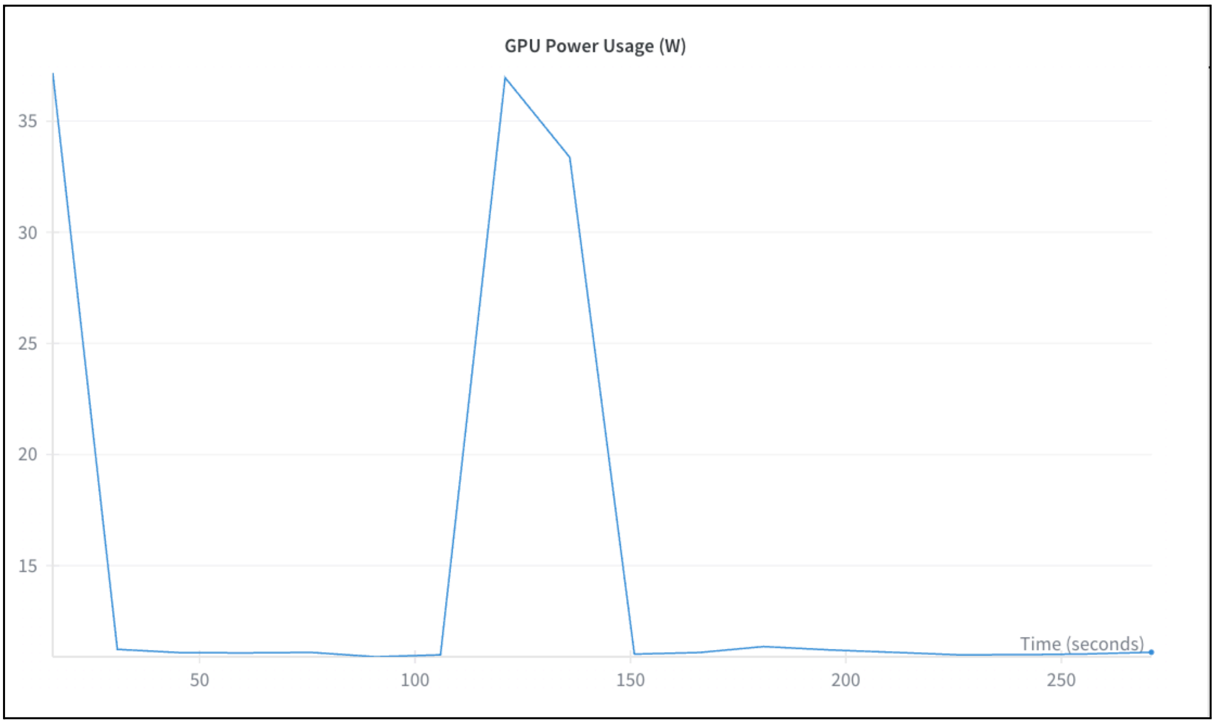
Comparison

Baseline Response:
FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FNstar, in

Fine-tuned Response:
SCAR-H assault rifles

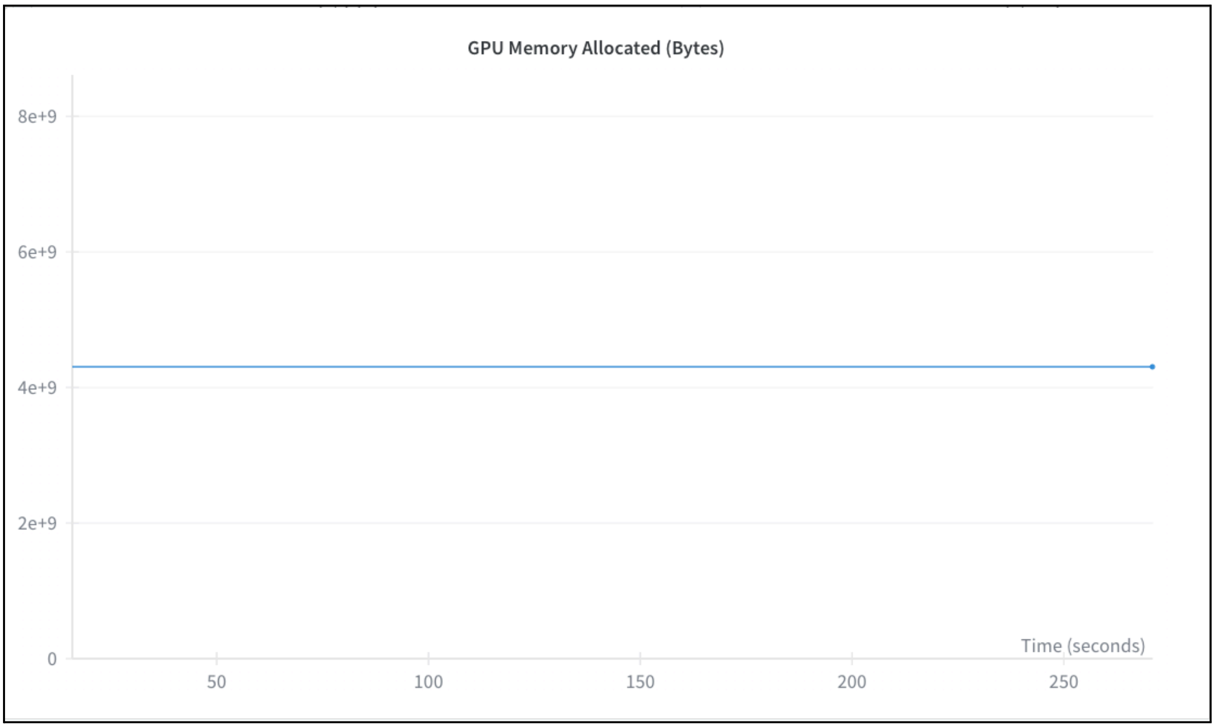
Based on the prompt above, the LoRA fine-tuned model demonstrates the ability to generate a more direct and accurate response compared to the baseline model, which produces an incorrect answer. This indicates that fine-tuning plays a crucial role in enhancing the overall quality of the model by improving its accuracy, contextual understanding, and relevance to the given prompt.

GPU Power Usage (W)



The GPU Power Usage (W) graph illustrates the GPU's power consumption over time (in seconds). Initially, the power usage is high but then gradually decreases, followed by a spike that reaches 37.15 W before dropping again and continuing to decrease steadily until the end of the process.

GPU Memory Allocated (Bytes)

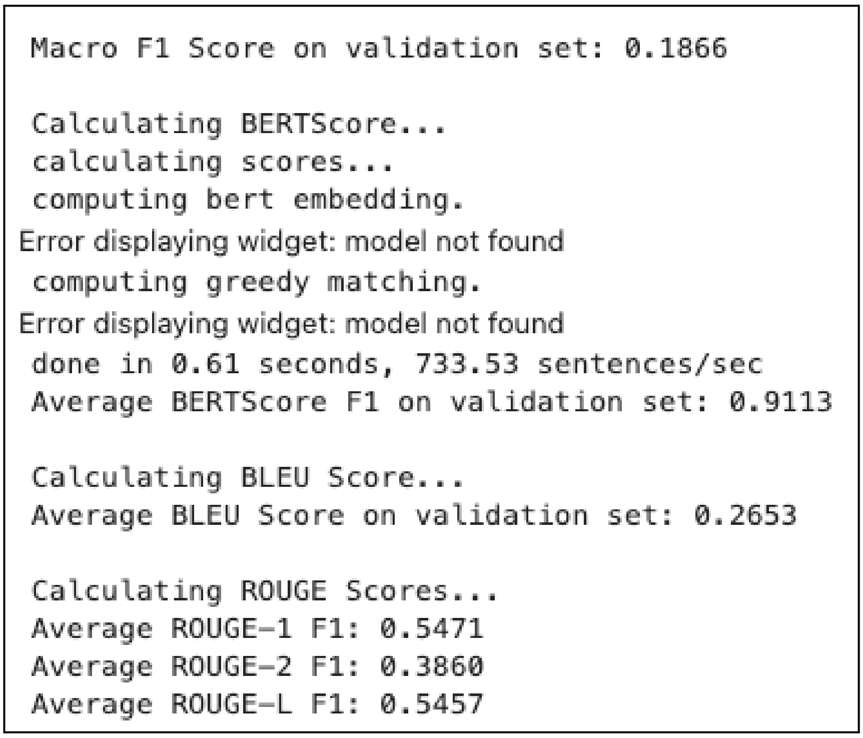


The graph above shows that the memory allocated for the LoRA fine-tuning process remains constant from the start to the end, with a usage of 4,304,994,304 bytes, which is approximately 4.30 GB. This indicates that the memory allocation is stable throughout the process, suggesting that LoRA fine-tuning is memory-efficient and does not require dynamic adjustments in GPU memory once the process has begun. Such stability is beneficial as it minimizes memory overhead and ensures consistent resource utilization during training

QLoRA

Sources: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/lygh83sf/overview

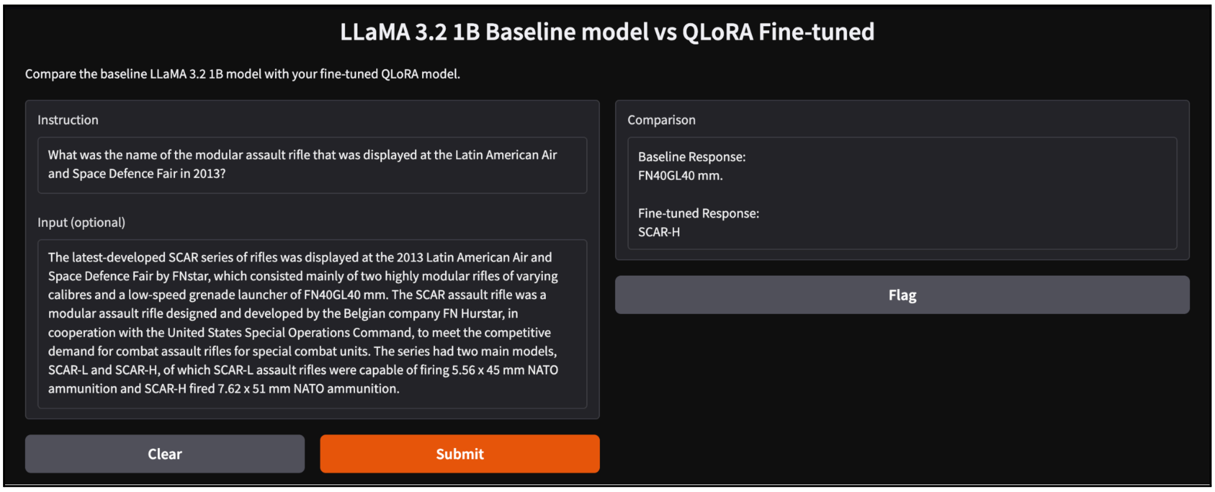
Metrics Evaluation



```
 Macro F1 Score on validation set: 0.1866

 Calculating BERTScore...
 calculating scores...
 computing bert embedding.
Error displaying widget: model not found
 computing greedy matching.
Error displaying widget: model not found
 done in 0.61 seconds, 733.53 sentences/sec
 Average BERTScore F1 on validation set: 0.9113

 Calculating BLEU Score...
 Average BLEU Score on validation set: 0.2653

 Calculating ROUGE Scores...
 Average ROUGE-1 F1: 0.5471
 Average ROUGE-2 F1: 0.3860
 Average ROUGE-L F1: 0.5457
```

After fine-tuning the model with QLoRA, there was a significant improvement across all evaluation metrics on the validation set compared to LoR fine-tuned model and baseline model. The Macro F1 Score increased to 0.1866, indicating better overall classification balance compared to the previous result. The BERTScore F1 also rose to 0.9113, showing stronger semantic similarity between the generated outputs and the reference texts. In terms of surface-level text matching, the BLEU Score improved to 0.2653, reflecting higher n-gram precision. Substantial gains were also observed in the ROUGE metrics, with ROUGE-1 F1 at 0.5471, ROUGE-2 F1 at 0.3860, and ROUGE-L F1 at 0.5457, all of which suggest that the model now captures more relevant words, phrases, and sequence structures from the target text. Overall, LoRA fine-tuning enhanced the model's ability to produce more accurate, fluent, and contextually aligned outputs.

Human Evaluation



LLaMA 3.2 1B Baseline model vs QLoRA Fine-tuned

Compare the baseline LLaMA 3.2 1B model with your fine-tuned QLoRA model.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

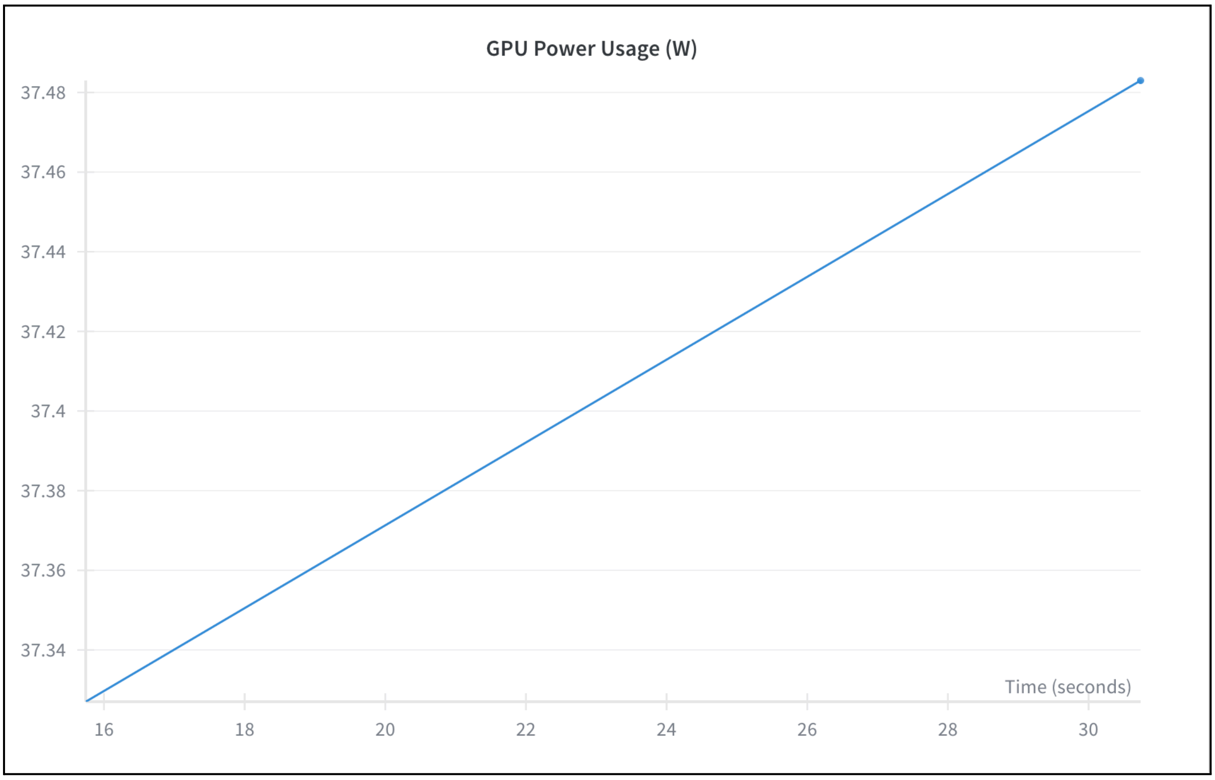
Comparison

Baseline Response:
FN40GL40 mm.

Fine-tuned Response:
SCAR-H

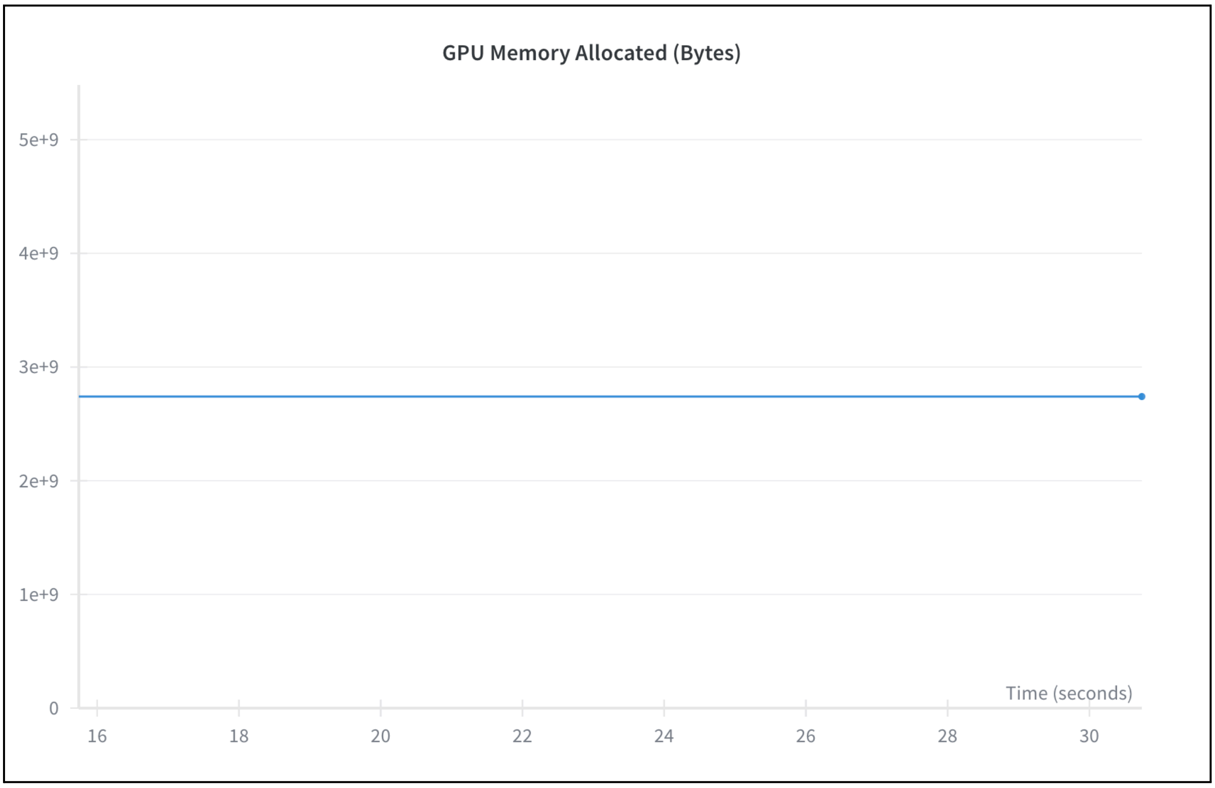
Based on the prompt above, the QLoRA fine-tuned model provides the correct answer of SCAR-H compared to the baseline model, which gives a wrong result. This shows that QLoRA fine-tuning improves the model's ability to generate accurate and contextually appropriate responses to the given prompt

GPU Power Usage (W)



The GPU Power Usage (W) graph illustrates the GPU's power consumption over time (in seconds). The power usage starts small and increases gradually until it reaches its peak at 37.483.

GPU Memory Allocated (Bytes)

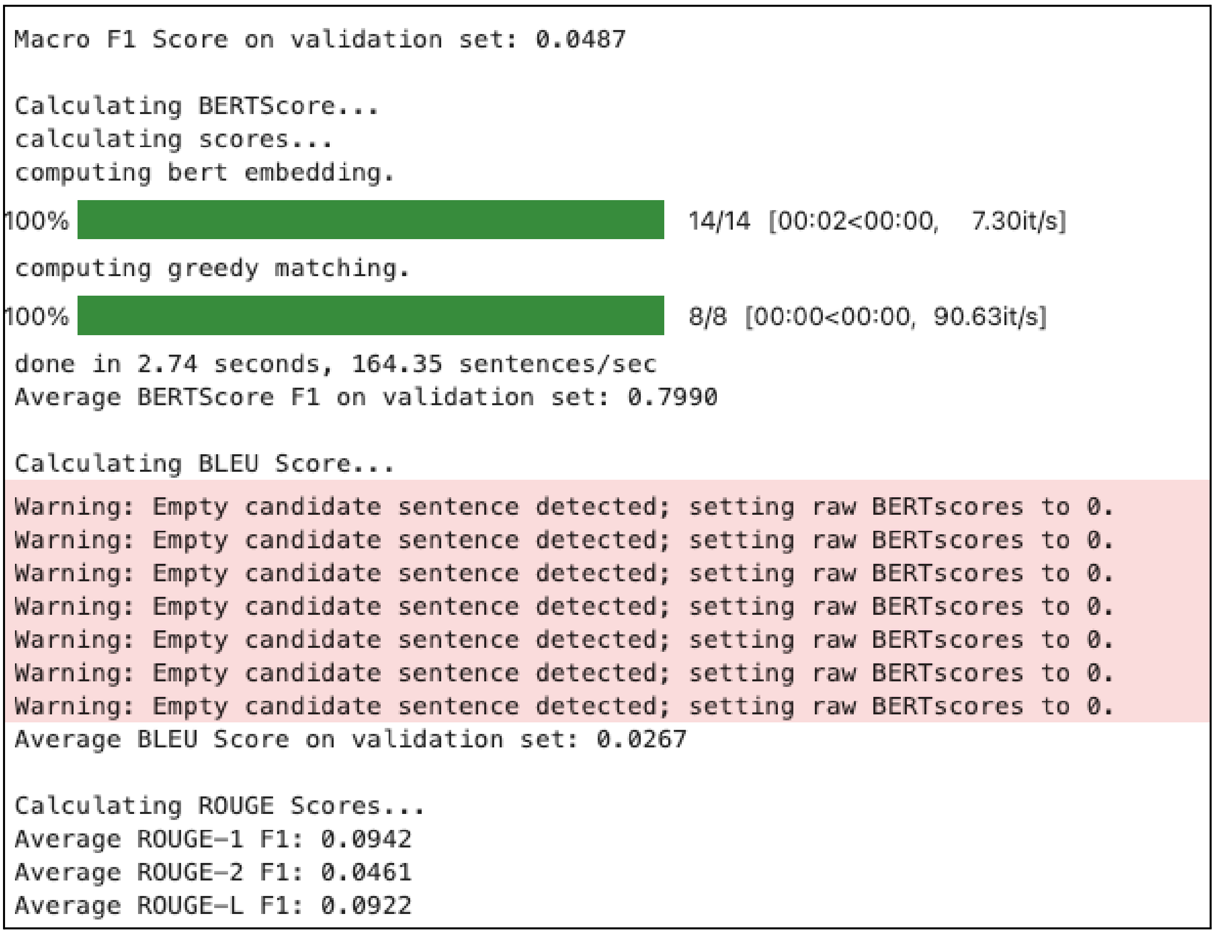


The graph above shows that the memory usage remains constant throughout the process, with a total of 2,740,518,912 bytes allocated, which is approximately 2.74 GB. This indicates that the process maintains a stable memory requirement from start to finish without significant fluctuations.

Top Layer

Sources :
https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/k52cbfpn/overview

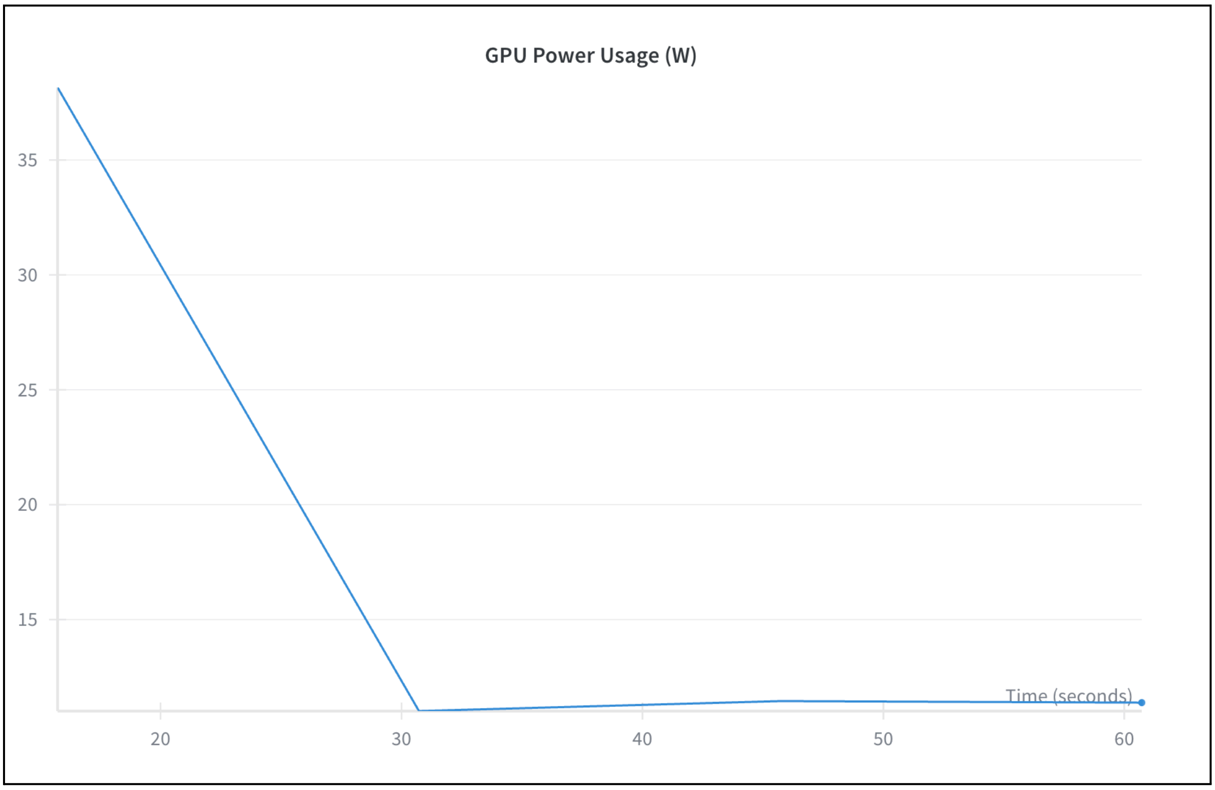
Metrics Evaluation



```
Macro F1 Score on validation set: 0.0487

Calculating BERTScore...
calculating scores...
computing bert embedding.
100%                                          14/14 [00:02<00:00,  7.30it/s]
computing greedy matching.
100%                                          8/8 [00:00<00:00, 90.63it/s]
done in 2.74 seconds, 164.35 sentences/sec
Average BERTScore F1 on validation set: 0.7990

Calculating BLEU Score...
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Average BLEU Score on validation set: 0.0267

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.0942
Average ROUGE-2 F1: 0.0461
Average ROUGE-L F1: 0.0922
```

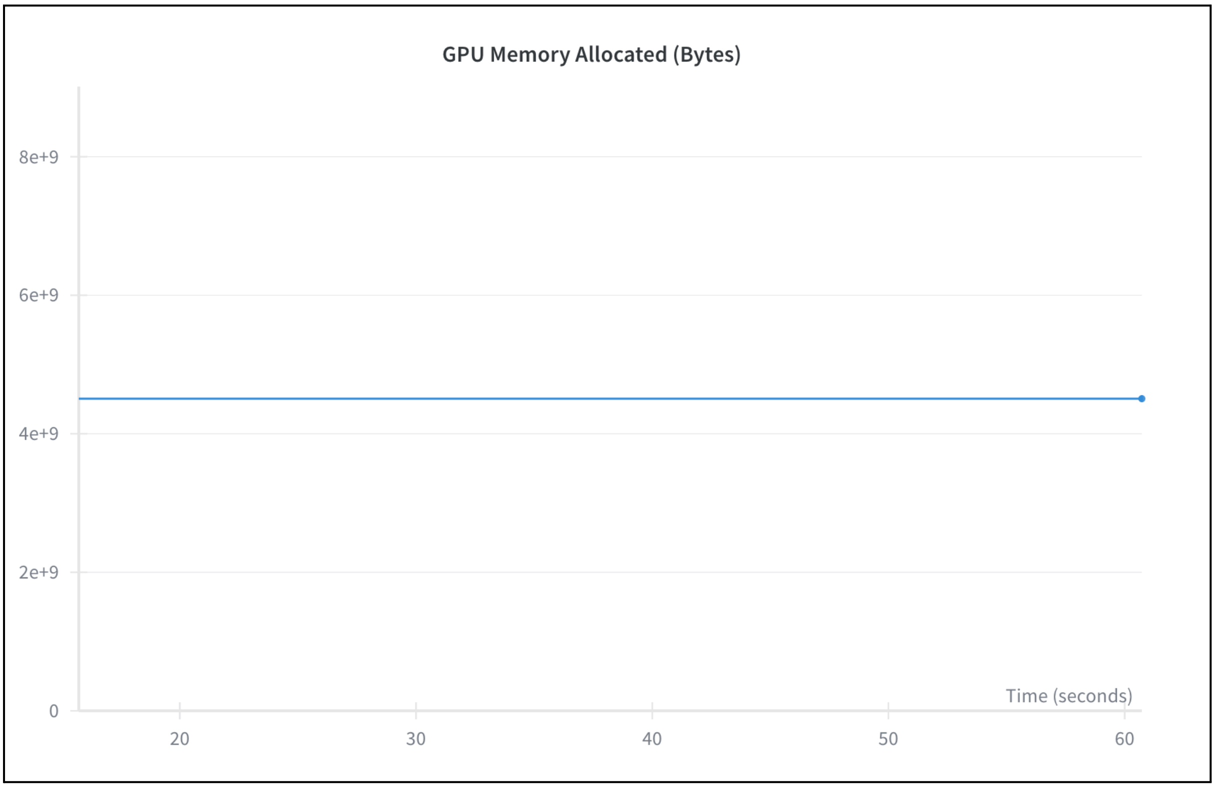
After fine-tuning using only the top layer, the model achieved a Macro F1 Score of 0.0487 on the validation set, indicating limited ability to correctly classify across all categories. The BERTScore F1 averaged at 0.7990, suggesting the model's generated responses share a relatively high semantic similarity with the reference answers. However, the BLEU Score was low at 0.0267, pointing to minimal overlap in exact wording. Similarly, the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores were 0.0942, 0.0461, and 0.0922 respectively, showing that while there is some overlap in n-grams and sequence structure, it remains quite limited, indicating the model still struggles with precise lexical reproduction despite retaining some semantic closeness.

GPU Power Usage (W)



The power usage started high, reaching its peak at 38.156 W before gradually decreasing over time. After the initial drop, the power consumption shows a steady yet slight increase, indicating that while the computational demand reduces after the peak, the GPU continues to handle smaller but consistent workloads throughout the process.

GPU Memory Allocated (Bytes)



The memory allocated during the process remains constant at 4,506,320,896 bytes, which is approximately 4.51 GB. This stable allocation indicates that the process does not require additional memory once it has started, reflecting consistent usage of GPU resources throughout the entire duration.

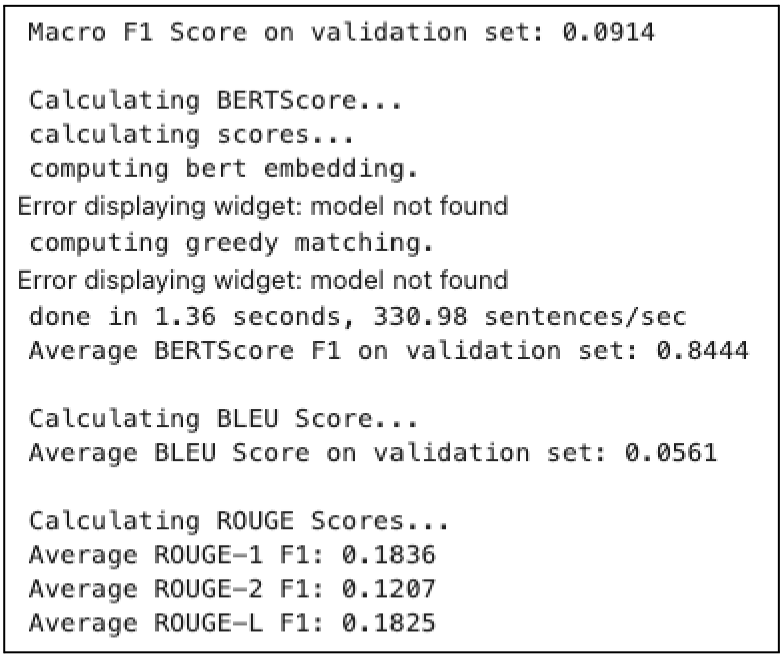
Table of comparison Llama 3.2 1B

Method	Trainable parameters (%)	Macro F1	BLEU	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L	GPU Power Usage (W)	GPU Memory Allocated (GB)
Baseline	-	0.0234	0.0148	0.8175	0.0663	0.0263	0.0644	68.97	16.59
LoRA	0.069%	0.1825	0.2243	0.8979	0.4664	0.3123	0.4638	37.15	4.30
QLoRA	0.069%	0.1866	0.2653	0.9113	0.5471	0.3860	0.5457	37.48	2.74
TopLayer	4.92%	0.0487	0.0267	0.8167	0.0942	0.0461	0.0922	38.16	4.51

Llama 3.2 3B

Baseline model

https://wandb.ai/ahmadhakimiadnan-other/baseline-gpu-monitor/runs/p194w05u/overview



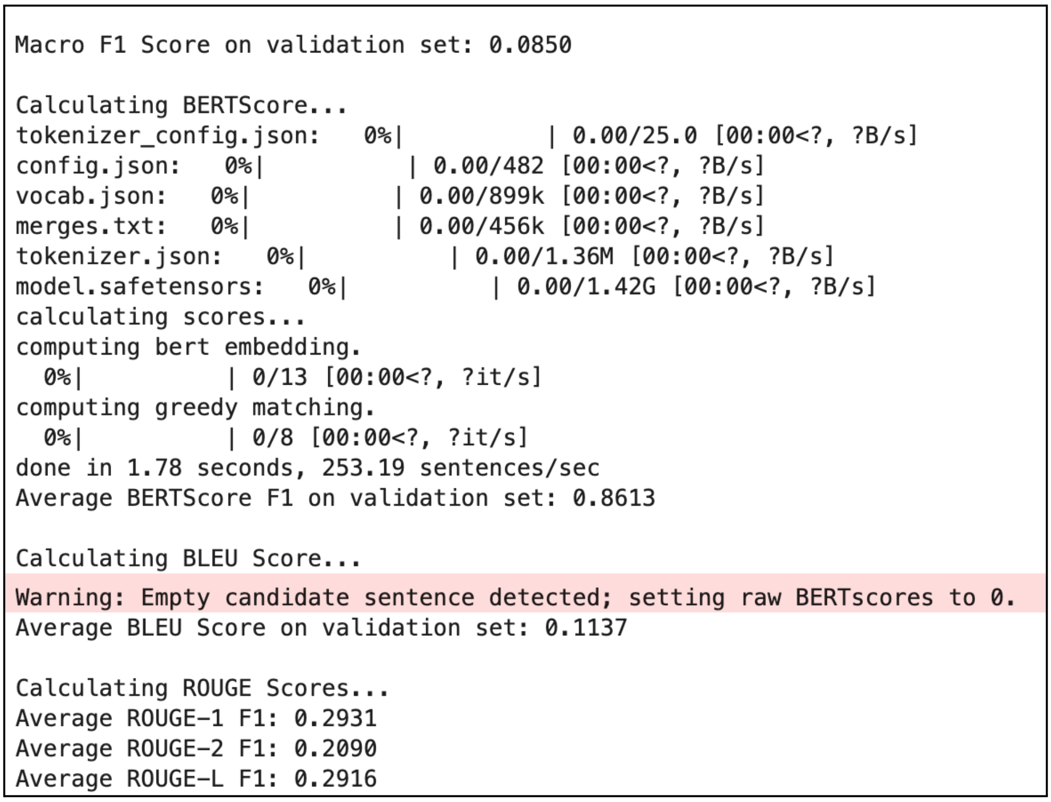
```
Macro F1 Score on validation set: 0.0914

Calculating BERTScore...
calculating scores...
computing bert embedding.
Error displaying widget: model not found
computing greedy matching.
Error displaying widget: model not found
done in 1.36 seconds, 330.98 sentences/sec
Average BERTScore F1 on validation set: 0.8444

Calculating BLEU Score...
Average BLEU Score on validation set: 0.0561

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.1836
Average ROUGE-2 F1: 0.1207
Average ROUGE-L F1: 0.1825
```

The evaluation results indicate that the LlaMA 3.2 3B baseline model performed poorly overall on the validation set across most metrics. The Macro F1 Score was 0.0914, suggesting the model struggled with balanced performance across all classes. In terms of semantic similarity, the BERTScore F1 was relatively higher at 0.8444, indicating the generated text had some degree of contextual overlap with the reference outputs. However, the BLEU Score was only 0.0561, reflecting very low n-gram precision. Similarly, the ROUGE metrics showed weak performance, with ROUGE-1 F1 at 0.1836, ROUGE-2 F1 at 0.1207, and ROUGE-L F1 at 0.1825, meaning the model failed to capture much lexical or sequential similarity to the target text. Overall, while the semantic similarity metric (BERTScore) appears acceptable, the low F1, BLEU, and ROUGE scores point to limited accuracy and fluency in the generated outputs.

LoRA

Sources:
https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/okn0etmh/overview

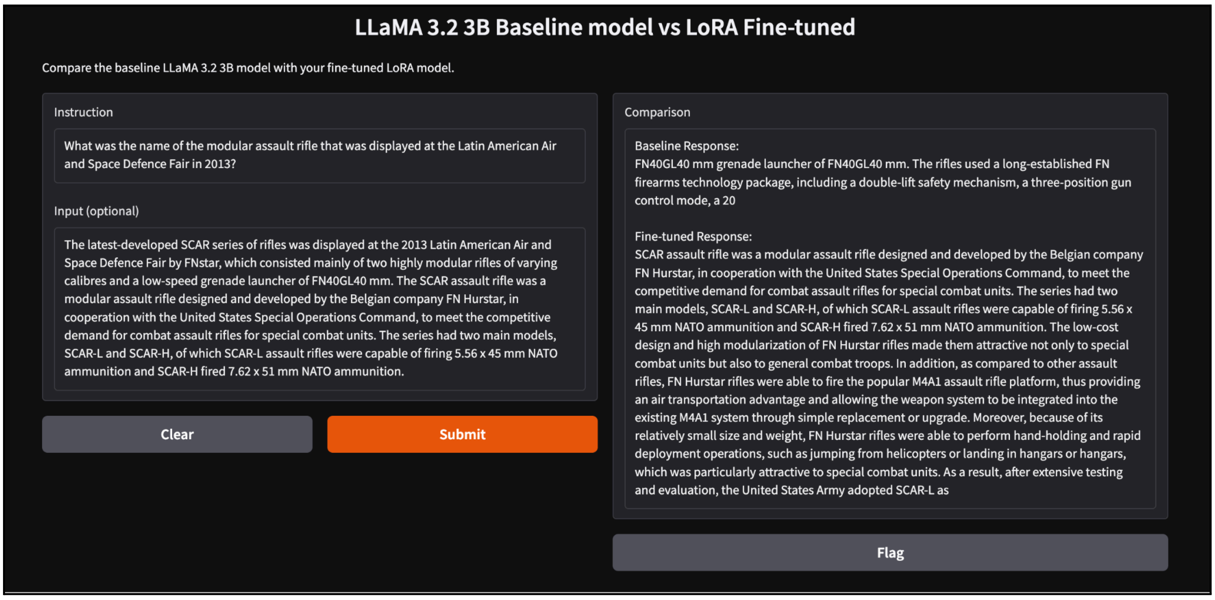
Metrics Evaluation



```
Macro F1 Score on validation set: 0.0850

Calculating BERTScore...
tokenizer_config.json:   0%|          | 0.00/25.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/482 [00:00<?, ?B/s]
vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]
merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/1.36M [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/1.42G [00:00<?, ?B/s]
calculating scores...
computing bert embedding.
  0%|          | 0/13 [00:00<?, ?it/s]
computing greedy matching.
  0%|          | 0/8 [00:00<?, ?it/s]
done in 1.78 seconds, 253.19 sentences/sec
Average BERTScore F1 on validation set: 0.8613

Calculating BLEU Score...
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Average BLEU Score on validation set: 0.1137

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.2931
Average ROUGE-2 F1: 0.2090
Average ROUGE-L F1: 0.2916
```

After fine-tuning the model with LoRA, there was a significant improvement across all evaluation metrics on the validation set. The Macro F1 Score increased to 0.0850, indicating better overall classification balance compared to the previous result. The BERTScore F1 also rose to 0.8613, showing stronger semantic similarity between the generated outputs and the reference texts. In terms of surface-level text matching, the BLEU Score improved to 0.1137, reflecting higher n-gram precision. Substantial gains were also observed in the ROUGE metrics, with ROUGE-1 F1 at 0.2931, ROUGE-2 F1 at 0.2090, and ROUGE-L F1 at 0.2916, all of which suggest that the model now captures more relevant words, phrases, and sequence structures from the target text. Overall, LoRA fine-tuning enhanced the model's ability to produce more accurate, fluent, and contextually aligned outputs.

Human Evaluation



LLaMA 3.2 3B Baseline model vs LoRA Fine-tuned

Compare the baseline LLaMA 3.2 3B model with your fine-tuned LoRA model.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

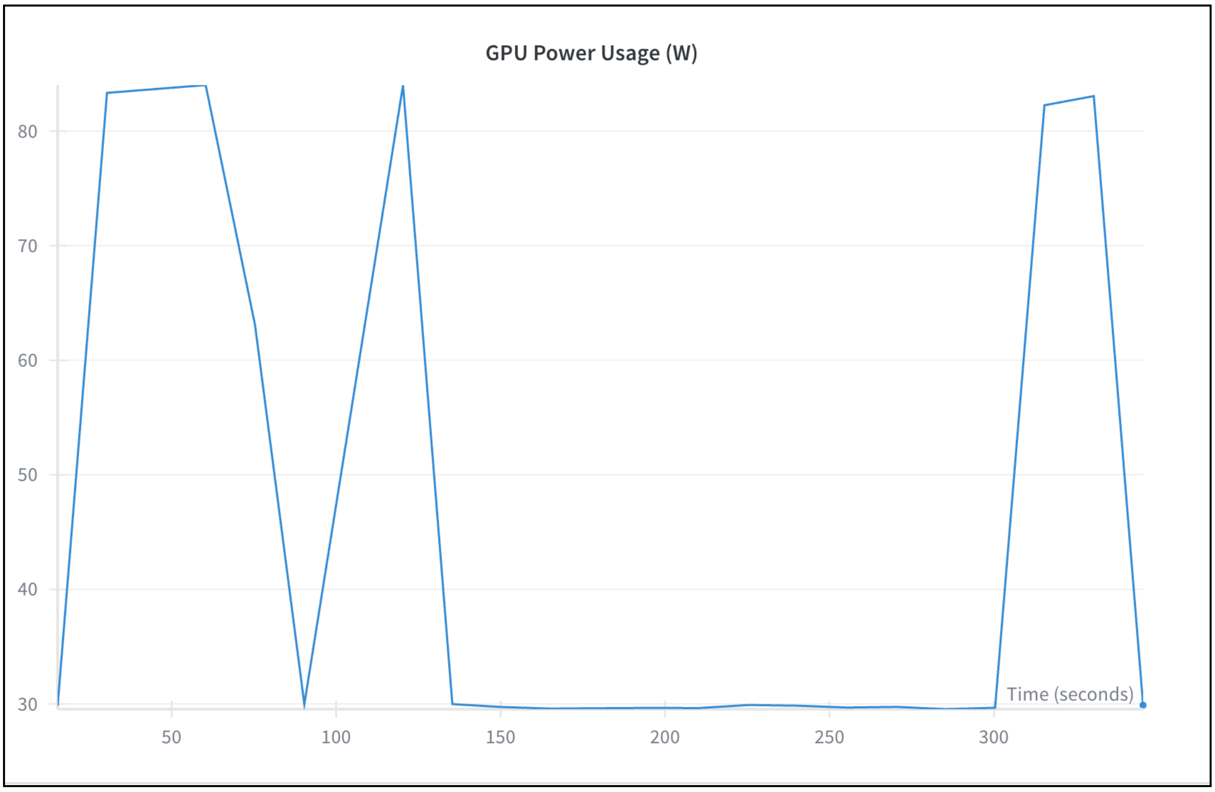
Comparison

Baseline Response:
FN40GL40 mm grenade launcher of FN40GL40 mm. The rifles used a long-established FN firearms technology package, including a double-lift safety mechanism, a three-position gun control mode, a 20

Fine-tuned Response:
SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. The low-cost design and high modularization of FN Hurstar rifles made them attractive not only to special combat units but also to general combat troops. In addition, as compared to other assault rifles, FN Hurstar rifles were able to fire the popular M4A1 assault rifle platform, thus providing an air transportation advantage and allowing the weapon system to be integrated into the existing M4A1 system through simple replacement or upgrade. Moreover, because of its relatively small size and weight, FN Hurstar rifles were able to perform hand-holding and rapid deployment operations, such as jumping from helicopters or landing in hangars or hangars, which was particularly attractive to special combat units. As a result, after extensive testing and evaluation, the United States Army adopted SCAR-L as

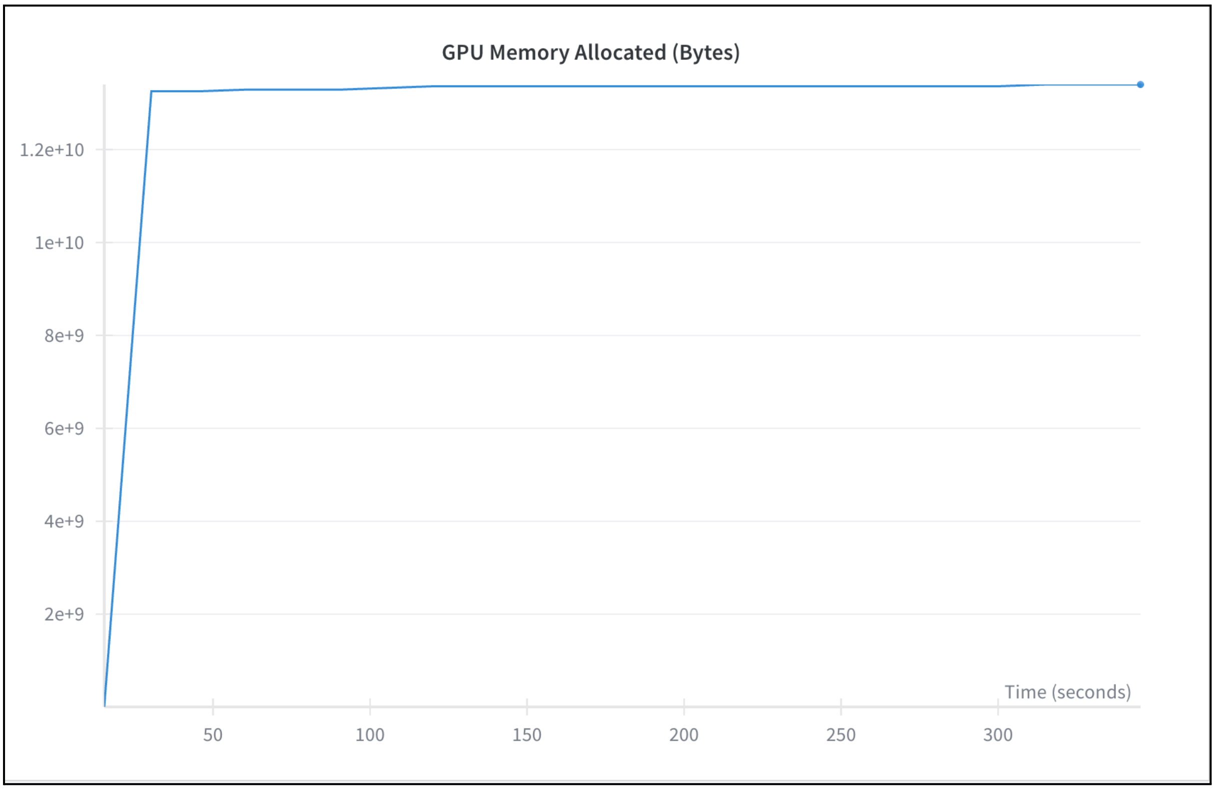
In contrast to the fine-tuned response, the baseline model provides the incorrect response based on the above instruction. However, the refined model's response is unnecessarily long and redundant.

GPU Power Usage (W)



The GPU power usage during the fine-tuning process showed noticeable fluctuations throughout the training. Initially, the power consumption increased rapidly as the model and data were loaded onto the GPU. After this initial spike, the usage stabilized briefly but continued to exhibit small oscillations, reflecting the varying computational demands of different training steps. The highest observed peak in GPU power consumption reached 84.009 W, indicating the moments of maximum computational load, likely corresponding to intensive forward and backward passes through the model. Following this peak, the power usage gradually decreased and remained variable until the end of the training process, illustrating how the GPU dynamically adapts its power draw in response to workload intensity.

GPU Memory Allocated (Bytes)

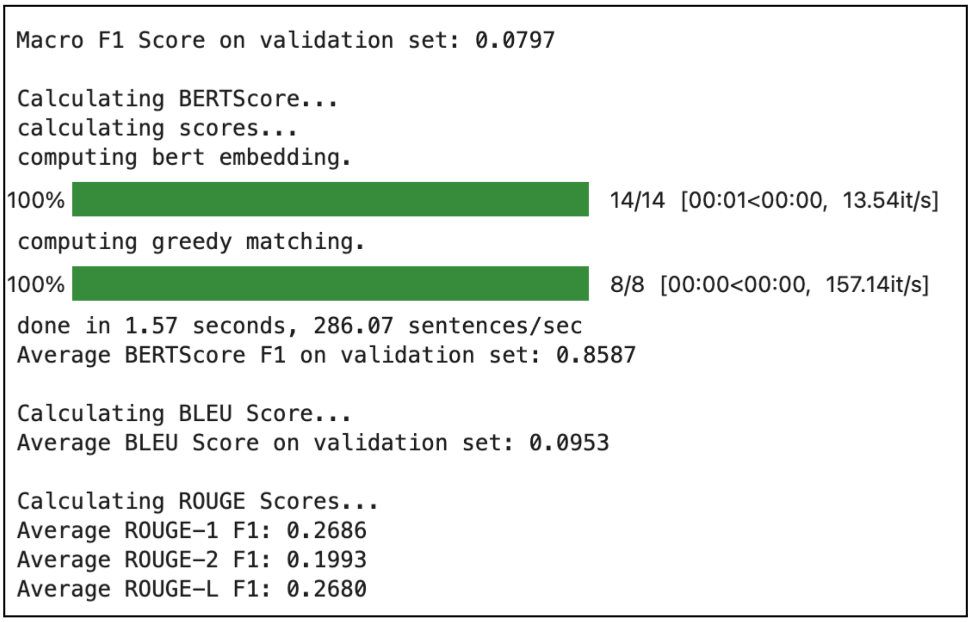


The GPU memory allocated during the training process gradually increased as the model and data were loaded and computations progressed. This increase continued steadily until it reached a peak of 13,253,476,352 bytes, approximately 13.25 GB. After reaching this maximum allocation, the memory usage remained constant for the remainder of the process, indicating that the GPU had fully reserved the required memory for the model, optimizer states, and intermediate computations. The plateau in memory usage suggests that there were no additional dynamic allocations beyond this point, and the GPU resources were being efficiently utilized throughout the rest of the training.

QLoRA

Sources: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/2x87fal6/overview

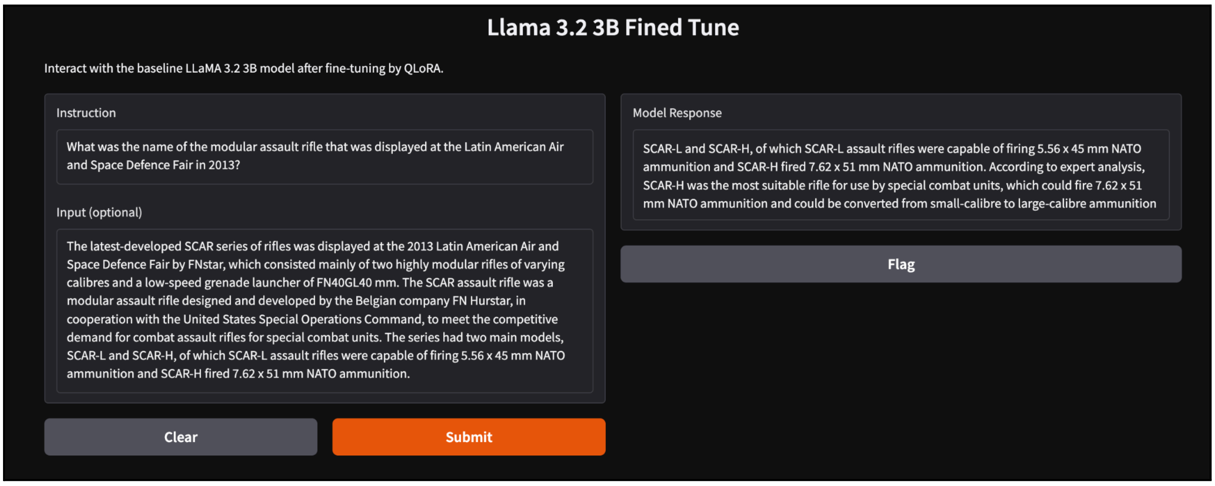
Metrics Evaluation



```
Macro F1 Score on validation set: 0.0797

Calculating BERTScore...
calculating scores...
computing bert embedding.
100% ████████████████████ 14/14 [00:01<00:00, 13.54it/s]
computing greedy matching.
100% ████████████████████ 8/8 [00:00<00:00, 157.14it/s]
done in 1.57 seconds, 286.07 sentences/sec
Average BERTScore F1 on validation set: 0.8587

Calculating BLEU Score...
Average BLEU Score on validation set: 0.0953

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.2686
Average ROUGE-2 F1: 0.1993
Average ROUGE-L F1: 0.2680
```

After fine-tuning the model with QLoRA, there was a significant improvement across all evaluation metrics on the validation set. The Macro F1 Score increased to 0.0797, showing lower result than LoRA fine-tuning indicating bad overall classification balance compared to the previous result. The BERTScore F1 also only 0.8587, showing stronger semantic similarity between the generated outputs and the reference texts compared to the baseline model. In terms of surface-level text matching, the BLEU Score improved to 0.0953, reflecting higher n-gram precision. Substantial gains were also observed in the ROUGE metrics, with ROUGE-1 F1 at 0.2686, ROUGE-2 F1 at 0.1993, and ROUGE-L F1 at 0.2680, slightly lower than LoRA fine-tuned result which suggest that the model now captures some relevant words, phrases, and sequence structures from the target text. Overall, QLoRA fine-tuning enhanced the model’s ability to produce more accurate, fluent, and contextually aligned outputs but still perform worserthan LoRA.

Human Evaluation



Llama 3.2 3B Fined Tune

Interact with the baseline LLaMA 3.2 3B model after fine-tuning by QLoRA.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

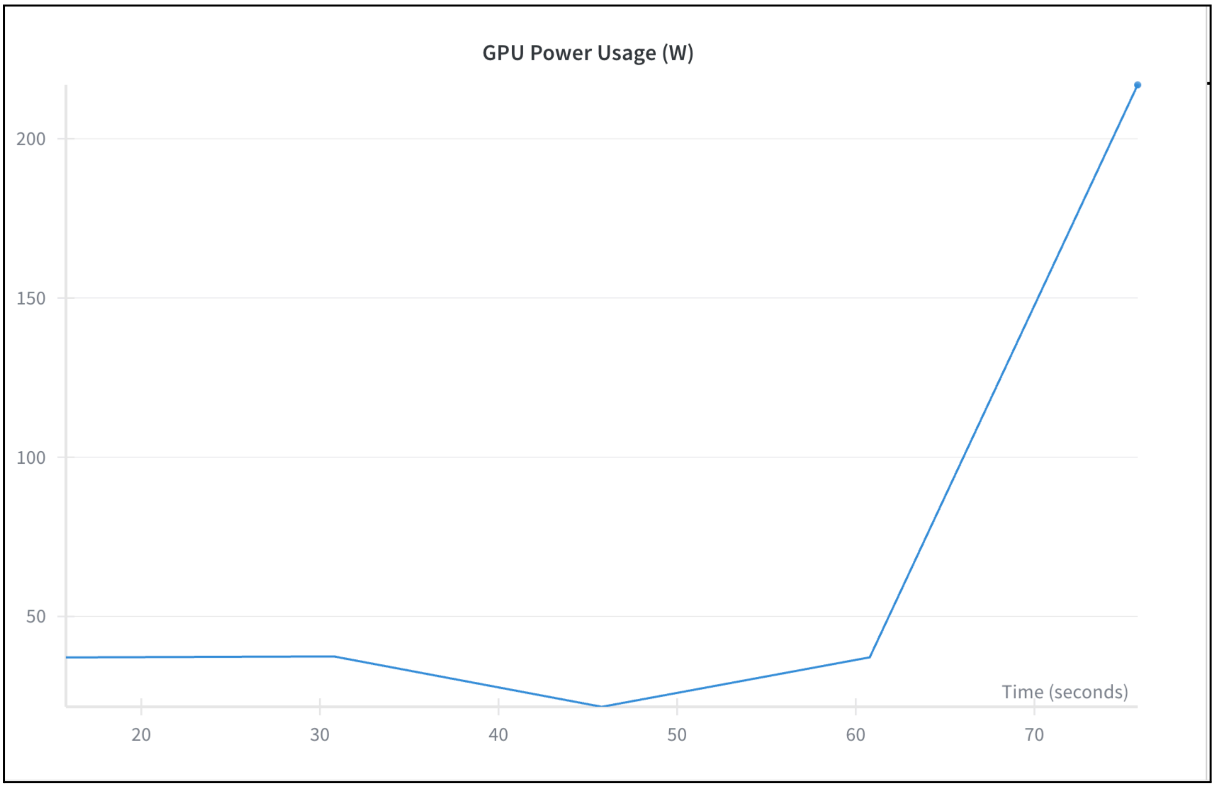
The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

Model Response

SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. According to expert analysis, SCAR-H was the most suitable rifle for use by special combat units, which could fire 7.62 x 51 mm NATO ammunition and could be converted from small-calibre to large-calibre ammunition

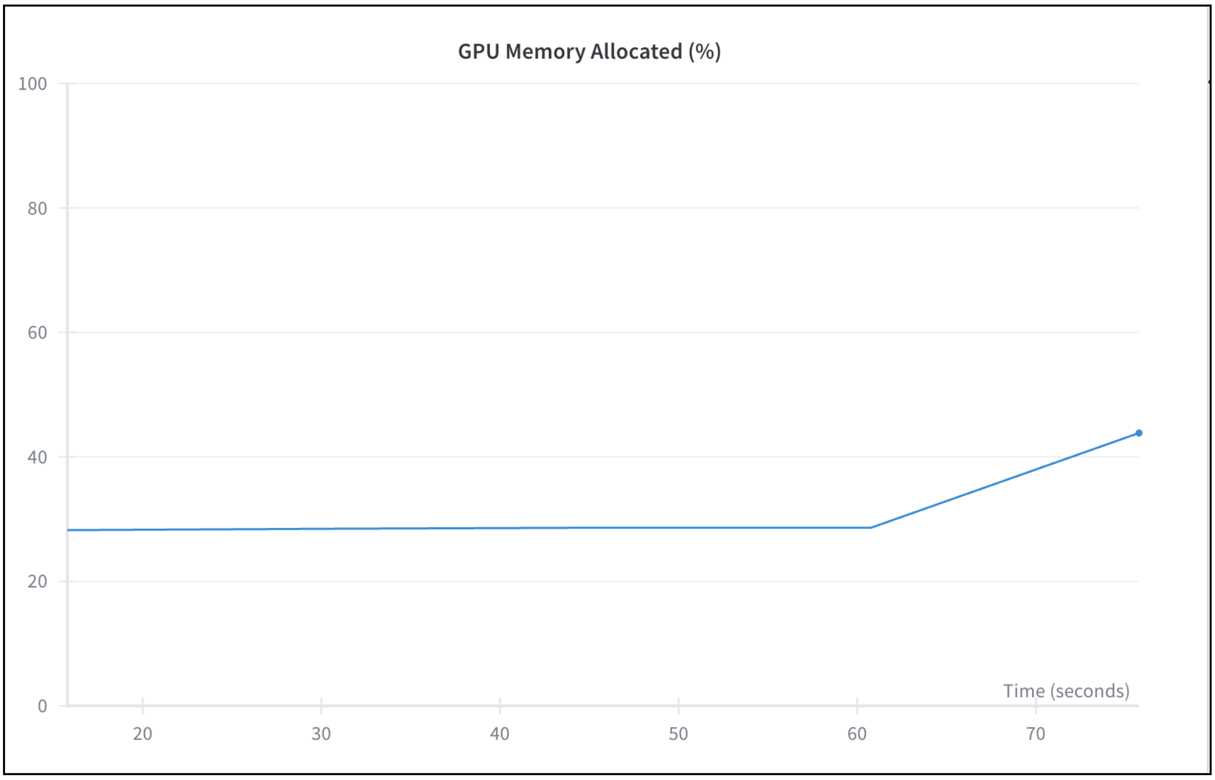
Model response answers correctly following the answer in the dataset, however the same thing happens as before that the answer is too long and redundant.

GPU Power Usage (W)



The GPU power usage starts low and gradually decreases, but at a certain point, it rises sharply until it reaches its peak at 200.10 W. This is the highest power consumption compared to any of the other methods. This is because QLoRA introduces additional low-rank adaptation matrices during training, which increases the number of active computations in the attention layers. As a result, the model requires more processing power, leading to higher energy consumption despite being more parameter-efficient than full fine-tuning.

GPU Memory Allocated (Bytes)

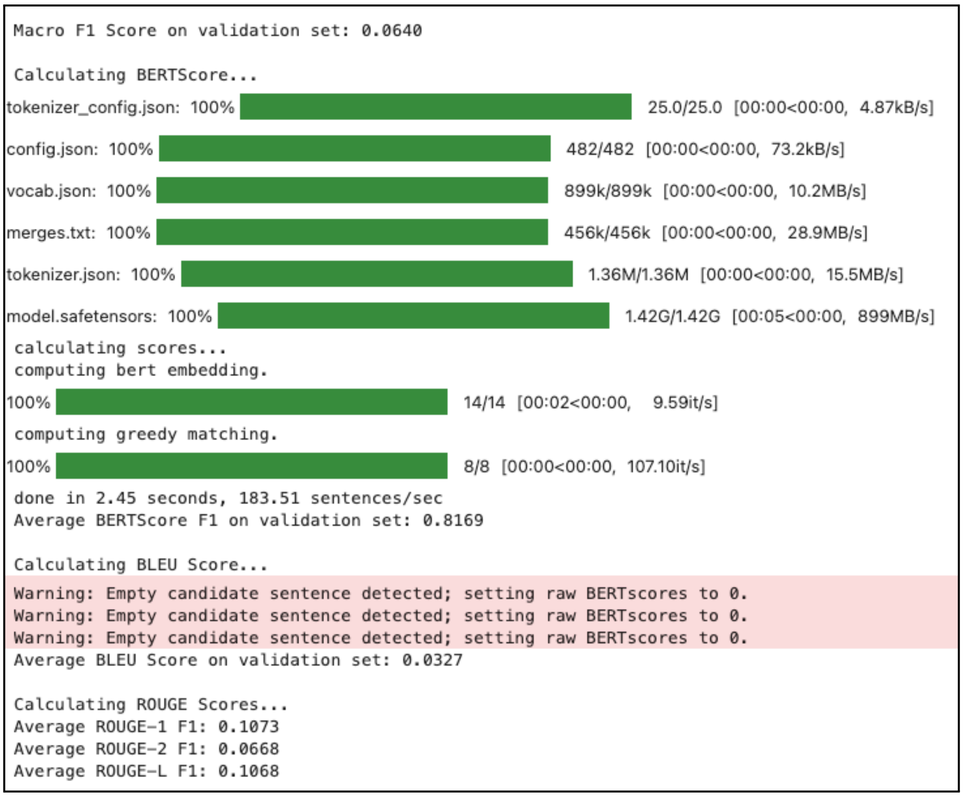


During the early stages of training, the memory allocation remains consistent at around 3.5 GB before gradually increasing to a peak of 4.38 GB. This demonstrates that QLoRA requires the least amount of memory throughout the training process, proving its suitability even when working with limited hardware resources

Top Layer

Sources: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/kubb1s2i/overview

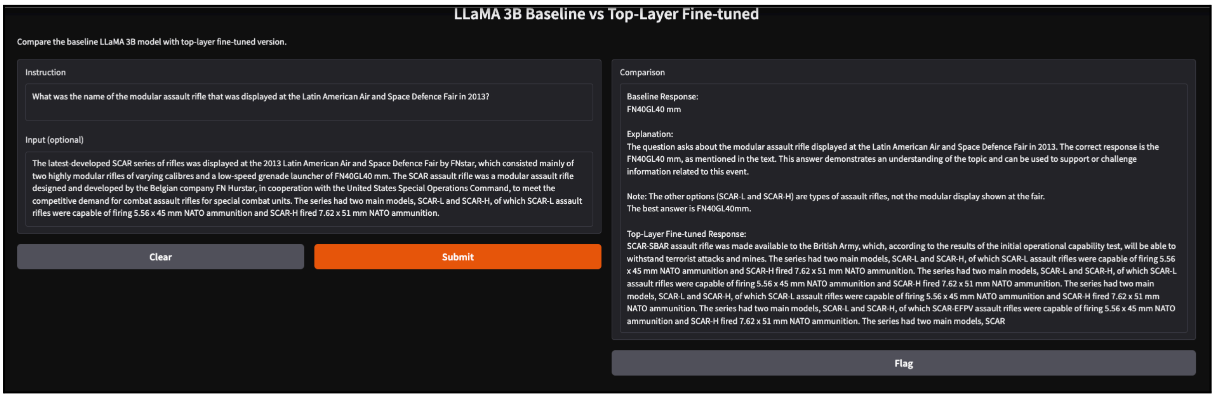
Metrics Evaluation



```
Macro F1 Score on validation set: 0.0640

Calculating BERTScore...
tokenizer_config.json: 100%  25.0/25.0 [00:00<00:00, 4.87kB/s]
config.json: 100%  482/482 [00:00<00:00, 73.2kB/s]
vocab.json: 100%  899k/899k [00:00<00:00, 10.2MB/s]
merges.txt: 100%  456k/456k [00:00<00:00, 28.9MB/s]
tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 15.5MB/s]
model.safetensors: 100%  1.42G/1.42G [00:05<00:00, 899MB/s]
calculating scores...
computing bert embedding.
100%  14/14 [00:02<00:00,  9.59it/s]
computing greedy matching.
100%  8/8 [00:00<00:00, 107.10it/s]
done in 2.45 seconds, 183.51 sentences/sec
Average BERTScore F1 on validation set: 0.8169

Calculating BLEU Score...
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Average BLEU Score on validation set: 0.0327

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.1073
Average ROUGE-2 F1: 0.0668
Average ROUGE-L F1: 0.1068
```

After fine-tuning using only the top layer, the model achieved a Macro F1 Score of 0.0640 on the validation set, indicating limited ability to correctly classify across all categories. The BERTScore F1 averaged at 0.8169, suggesting the model's generated responses share a relatively high semantic similarity with the reference answers. However, the BLEU Score was low at 0.0327, pointing to minimal overlap in exact wording. Similarly, the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores were 0.1073, 0.0668, and 0.1068 respectively, showing that while there is some overlap in n-grams and sequence structure, it remains quite limited, indicating the model still struggles with precise lexical reproduction despite retaining some semantic closeness.

Human Evaluation



LLaMA 3B Baseline vs Top-Layer Fine-tuned

Compare the baseline LLaMA 3B model with top-layer fine-tuned version.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

Comparison

Baseline Response:
FN40GL40 mm

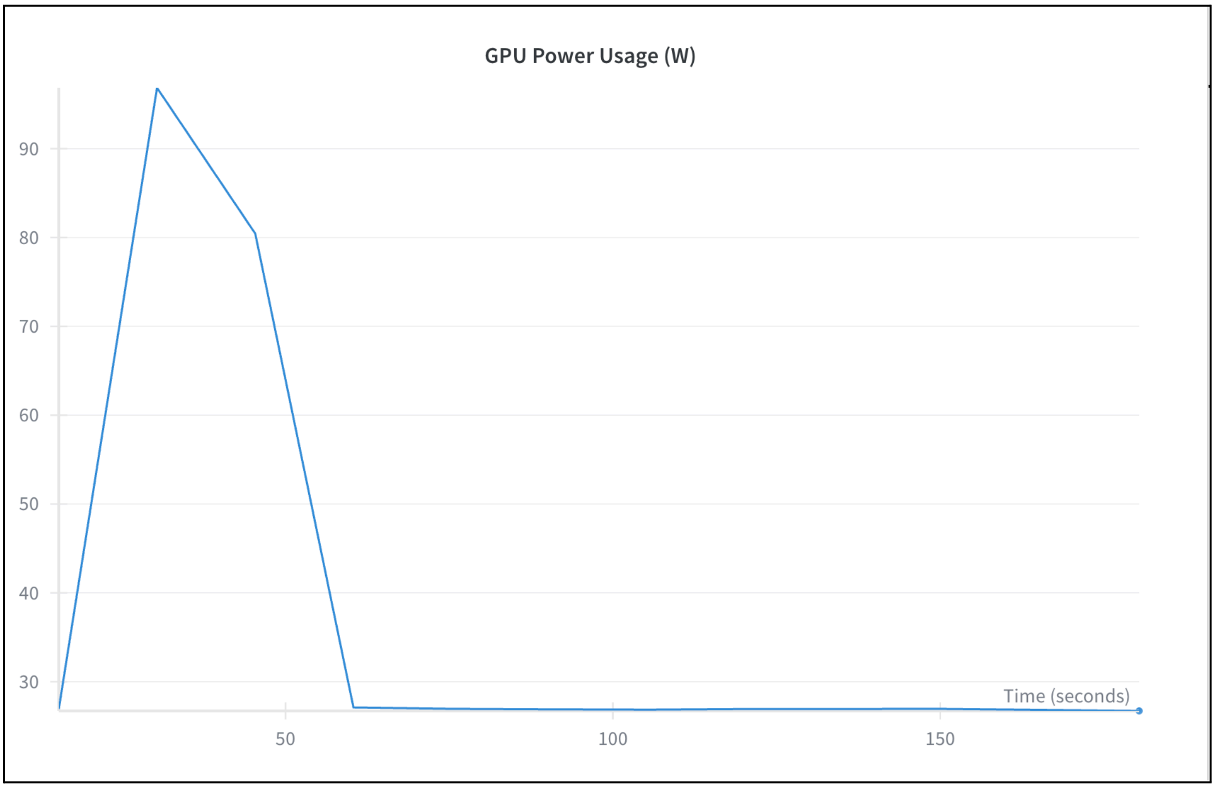
Explanation:
The question asks about the modular assault rifle displayed at the Latin American Air and Space Defence Fair in 2013. The correct response is the FN40GL40 mm, as mentioned in the text. This answer demonstrates an understanding of the topic and can be used to support or challenge information related to this event.

Note: The other options (SCAR-L and SCAR-H) are types of assault rifles, not the modular display shown at the fair.
The best answer is FN40GL40mm.

Top-Layer Fine-tuned Response:
SCAR-SBAR assault rifle was made available to the British Army, which, according to the results of the initial operational capability test, will be able to withstand terrorist attacks and mines. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. The series had two main models, SCAR-L and SCAR-H, of which SCAR-EFPV assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition. The series had two main models, SCAR

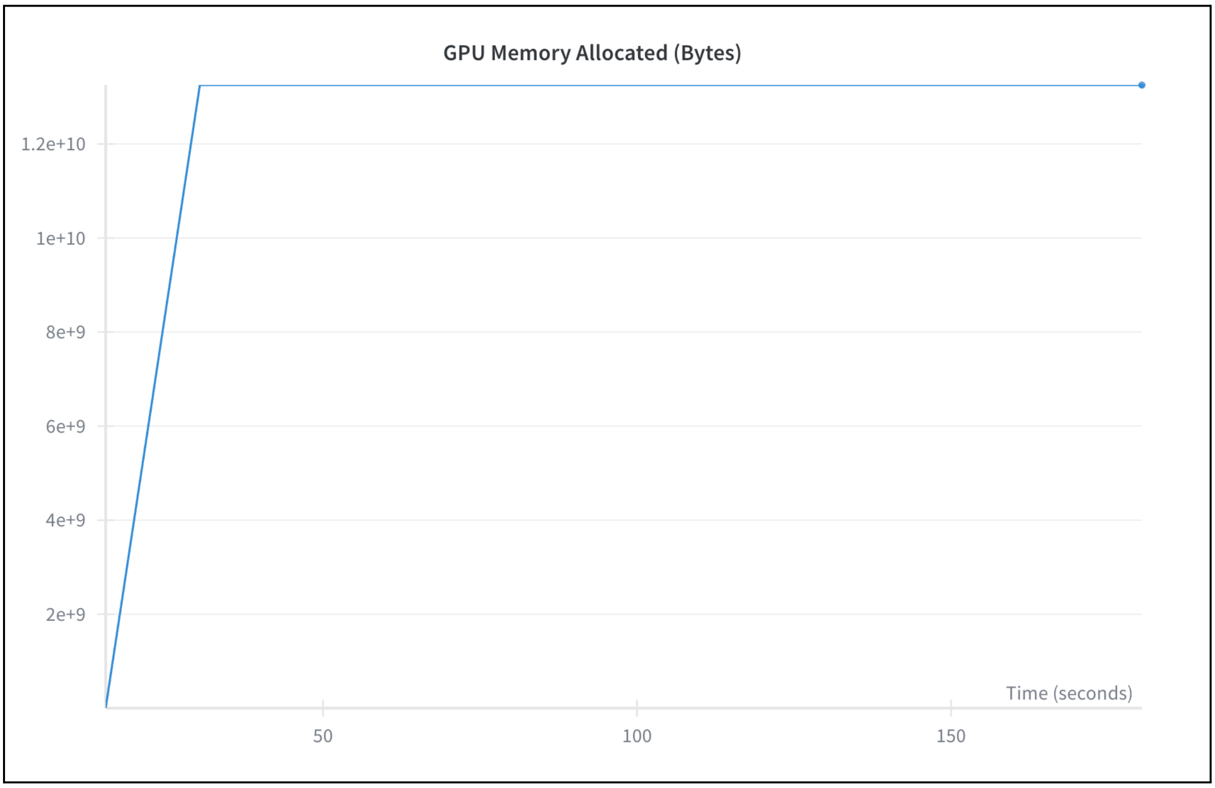
Based on the prompt above, the fine-tuned model gives a correct answer compared to the baseline model. This indicates that the fine-tuning process successfully enhanced the model’s understanding of the dataset, allowing it to generate more accurate and contextually appropriate responses. It shows that even though the overall evaluation metrics are relatively low, the fine-tuned model is capable of learning domain-specific distinctions and producing correct outputs where the baseline model struggled, highlighting the effectiveness of fine-tuning in improving model performance for targeted tasks.

GPU Power Usage (W)



The power usage increases rapidly at the early stage of the training process, reaching a peak of 96.849 W before gradually decreasing and stabilizing at a constant 27 W. This indicates that the initial phase of Top Layer training demands higher computational power to adjust the low-rank adaptation matrices, after which the process becomes more stable and energy-efficient.

GPU Memory Allocated (Bytes)



The memory allocated for the Top Layer method is the highest among all approaches, reaching 20.25 GB. Once it reaches this level, the allocation remains constant throughout most of the training process. This indicates that the Top Layer method is highly memory-intensive, requiring substantially more resources compared to other fine-tuning techniques such as QLoRA. As a result, it is less suitable for setups with limited hardware capacity, even though it may still provide competitive performance in terms of model accuracy

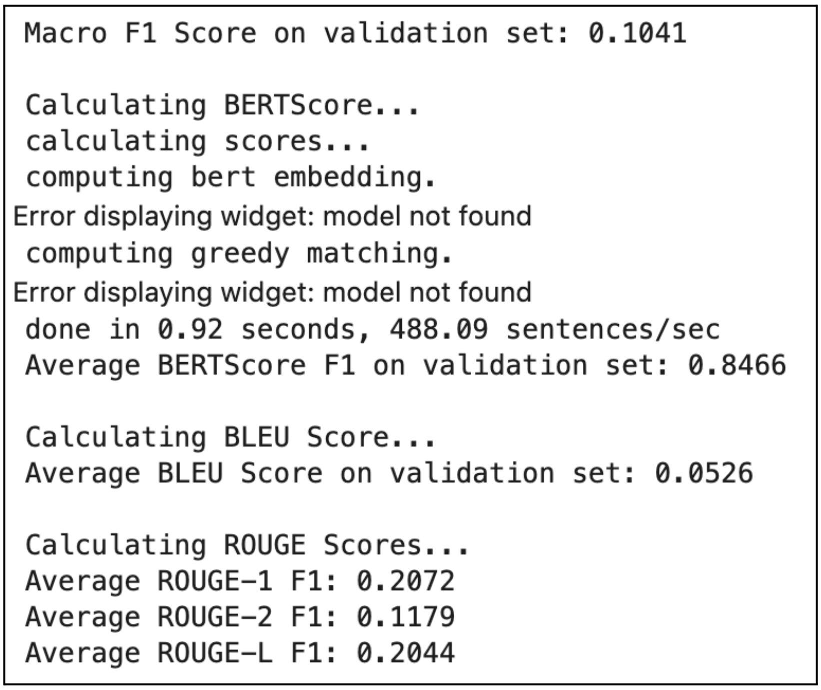
Table of comparison for Llama 3.2 3B

Method	Trainable parameters(%)	Macro F1	BLEU	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L	GPU Power Usage (W)	GPU Memory Allocated (GB)
Baseline	-	0.0914	0.0561	0.8444	0.1836	0.1207	0.1825	66.70	22.79
LoRA	0.2848	0.0950	0.1137	0.8613	0.2931	0.2090	0.2916	84.009	13.40
QLoRA	0.2848	0.0927	0.0953	0.8587	0.2686	0.1993	0.2680	84.108	4.38
TopLayer	31.10	0.0640	0.0327	0.8169	0.1073	0.0668	0.1068	96.849	20.25

Mistral 7B

Baseline model

https://wandb.ai/ahmadhakimiadnan-other/baseline-gpu-monitor/runs/ofkhd46i/overview



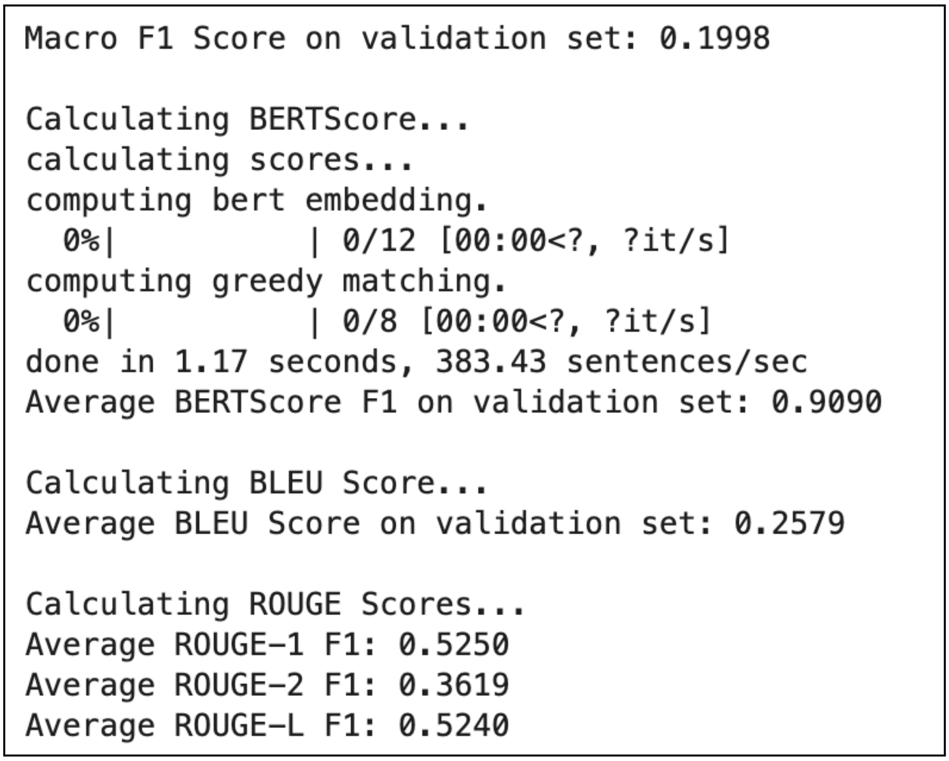
```
Macro F1 Score on validation set: 0.1041

Calculating BERTScore...
calculating scores...
computing bert embedding.
Error displaying widget: model not found
computing greedy matching.
Error displaying widget: model not found
done in 0.92 seconds, 488.09 sentences/sec
Average BERTScore F1 on validation set: 0.8466

Calculating BLEU Score...
Average BLEU Score on validation set: 0.0526

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.2072
Average ROUGE-2 F1: 0.1179
Average ROUGE-L F1: 0.2044
```

The evaluation results show that the Mistral 7B's performance is better compared to previous Llama 3.2 1B and 3.2 3B across different metrics. The low Macro F1 score 0.104 indicates that the model shows weak capability in correctly predicting minority classes. Similarly, the BLEU score 0.0526 is quite low, suggesting poor alignment with reference sequences at the n-gram level. However, the relatively high BERTScore F1 0.8466 implies that the model generates outputs that are semantically close to the references, even if not exact matches. The ROUGE scores ROUGE-1: 0.2072, ROUGE-2: 0.1179, ROUGE-L: 0.2044 further highlight that while there is some lexical overlap, it remains limited. Overall, the results indicate that the model can capture semantic meaning reasonably well but lacks precision and diversity in producing accurate and class-balanced outputs.

LoRA

Source: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/v5no5a7s

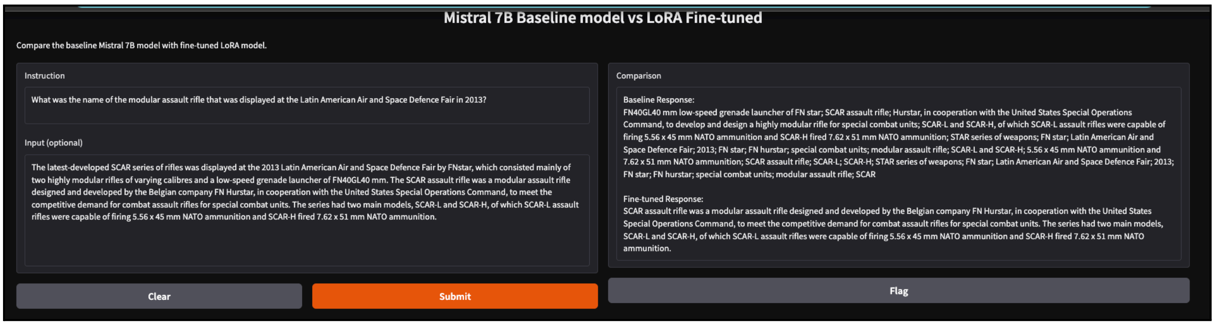
Metrics Evaluation



```
Macro F1 Score on validation set: 0.1998

Calculating BERTScore...
calculating scores...
computing bert embedding.
  0%|          | 0/12 [00:00<?, ?it/s]
computing greedy matching.
  0%|          | 0/8 [00:00<?, ?it/s]
done in 1.17 seconds, 383.43 sentences/sec
Average BERTScore F1 on validation set: 0.9090

Calculating BLEU Score...
Average BLEU Score on validation set: 0.2579

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.5250
Average ROUGE-2 F1: 0.3619
Average ROUGE-L F1: 0.5240
```

The LoRA fine-tuned model shows a noticeable improvement compared to the baseline. The Macro F1 score increases slightly to 0.1998, suggesting a better balance in handling tokens. More importantly, the semantic quality of the outputs is much stronger, as reflected in the higher BERTScore F1 of 0.9090. The BLEU score 0.2579 also improves significantly, indicating better alignment with reference sequences at the n-gram level. Likewise, the ROUGE scores ROUGE-1: 0.5250, ROUGE-2: 0.3619, ROUGE-L: 0.5240 show substantial gains, suggesting stronger lexical overlap and improved fluency in generated outputs. Overall, the LoRA model demonstrates better semantic accuracy, lexical similarity, and sequence-level coherence, highlighting its effectiveness over the baseline approach.

Human Evaluation



Compare the baseline Mistral 7B model with fine-tuned LoRA model.

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

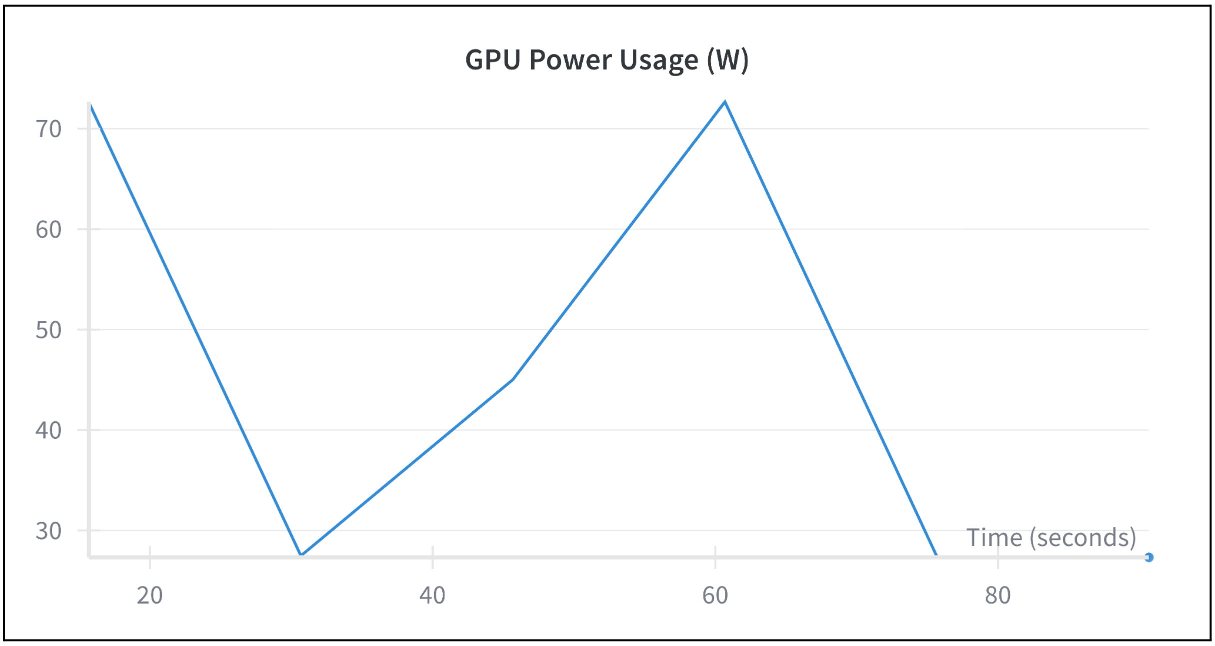
The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

Baseline Response:
FN40GL40 mm low-speed grenade launcher of FN star; SCAR assault rifle; Hurstar, in cooperation with the United States Special Operations Command, to develop and design a highly modular rifle for special combat units; SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition; STAR series of weapons; FN star; Latin American Air and Space Defence Fair; 2013; FN star; FN hurstar; special combat units; modular assault rifle; SCAR-L and SCAR-H; 5.56 x 45 mm NATO ammunition and 7.62 x 51 mm NATO ammunition; SCAR assault rifle; SCAR-L; SCAR-H; STAR series of weapons; FN star; Latin American Air and Space Defence Fair; 2013; FN star; FN hurstar; special combat units; modular assault rifle; SCAR

Fine-tuned Response:
SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

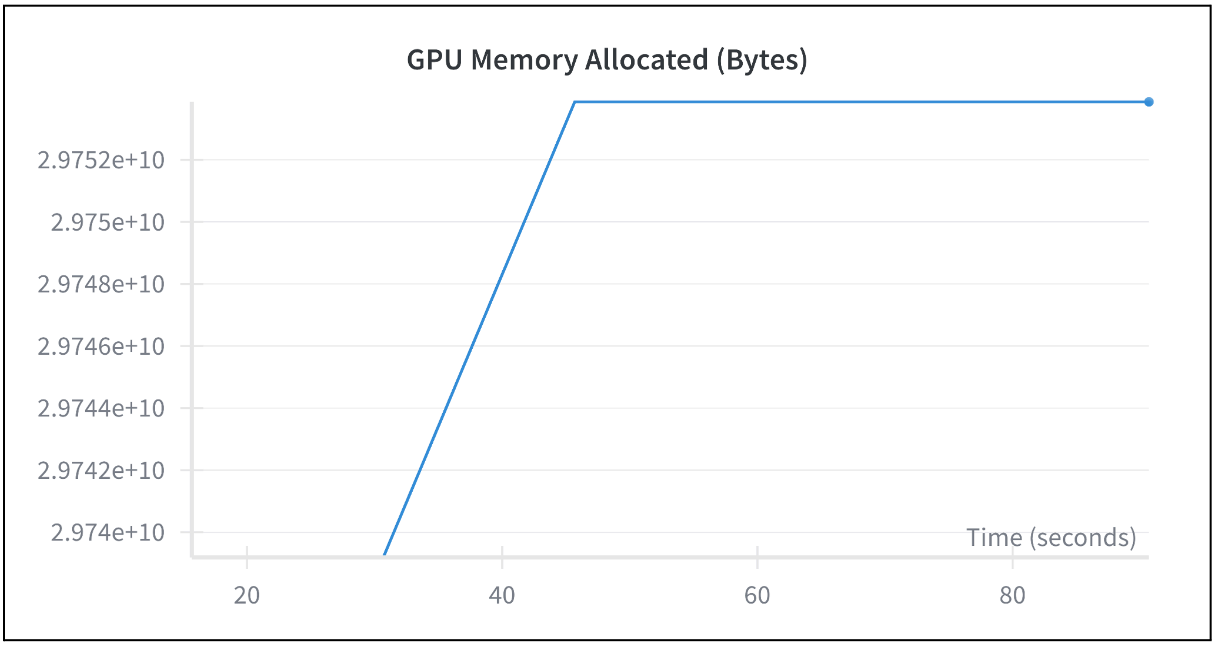
Based on the above prompt, the baseline model gives a better result compared to the baseline model. The model successfully answered SCAR assault without confusing it between the grenade launcher FN40GL40. This indicates that the model is able to distinguish between closely related terms within the same weapon family, showing its ability to capture semantic nuances in the text. However, despite this specific success case, the overall evaluation metrics reveal that the baseline model struggles with precision, lexical overlap, and class balance.

GPU Power Usage (W)



During the LoRA fine-tuning process, we can see from the diagram above that the GPU power usage fluctuates over time, reaching a peak of 72.672 W before gradually decreasing towards the end of training. This shows that the GPU was highly utilized during the early and middle stages of fine-tuning, where most of the computation and weight updates occurred, and then consumption dropped as the training stabilized and required less computational intensity. The downward trend towards the end reflects reduced workload, indicating that the model was converging and fewer resources were needed to complete the process efficiently.

GPU Memory Allocated (Bytes)

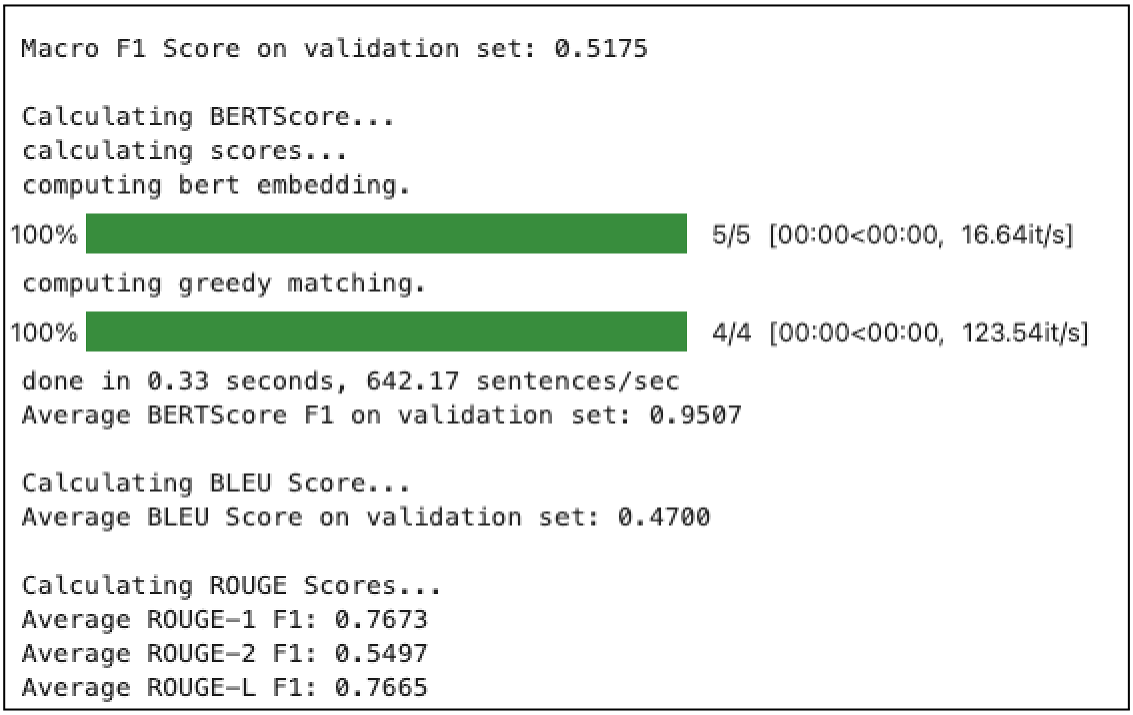


Based on the diagram above, the usage of GPU memory increased gradually until it reached its peak of **29753868288 bytes (29.75 GB)** and then remained constant until the end. This shows that the model gradually loaded the necessary parameters and training data into memory during the initial phase, after which the memory usage stabilized once all required resources were fully allocated. The constant usage until the end indicates that no additional memory was needed during the training process, suggesting efficient memory management and stable resource allocation throughout the fine-tuning.

QLoRA

Source: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/ktnsl6t7/overview

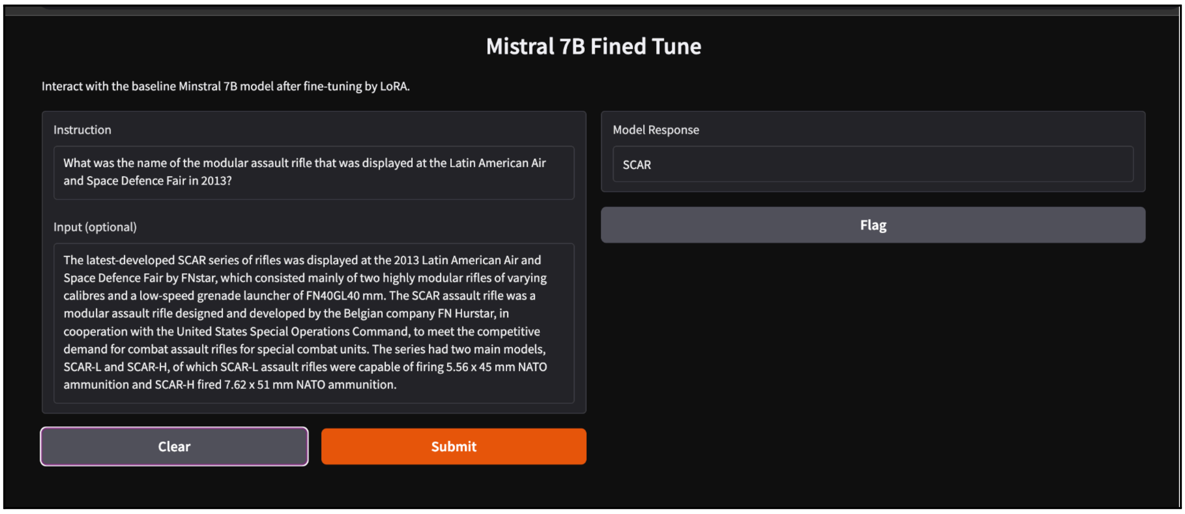
Metrics Evaluation



```
Macro F1 Score on validation set: 0.5175

Calculating BERTScore...
calculating scores...
computing bert embedding.
100%                                        5/5 [00:00<00:00, 16.64it/s]
computing greedy matching.
100%                                        4/4 [00:00<00:00, 123.54it/s]
done in 0.33 seconds, 642.17 sentences/sec
Average BERTScore F1 on validation set: 0.9507

Calculating BLEU Score...
Average BLEU Score on validation set: 0.4700

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.7673
Average ROUGE-2 F1: 0.5497
Average ROUGE-L F1: 0.7665
```

The fine-tuned QLoRA's model shows clear, across-the-board gains compared to the LoRA fine-tuned model. Macro F1 rises to 0.5175, indicating better class balance and improved recognition of minority classes. Semantic is strong with BERTScore F1 = 0.9507, while sequence and token-level overlap improve substantially BLEU = 0.4700, ROUGE-1/2/L F1 = 0.7673, 0.5497, 0.7665, suggesting outputs that are both closer in meaning and more textually aligned with references. In practice, this points to more accurate, fluent responses with fewer confusions between similar entities. Some headroom remains in Macro F1, implying occasional misses on rarer classes, but overall the model demonstrates more consistent and reliable performance than earlier runs.

Human Evaluation



Mistral 7B Fined Tune

Interact with the baseline Minstral 7B model after fine-tuning by LoRA.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

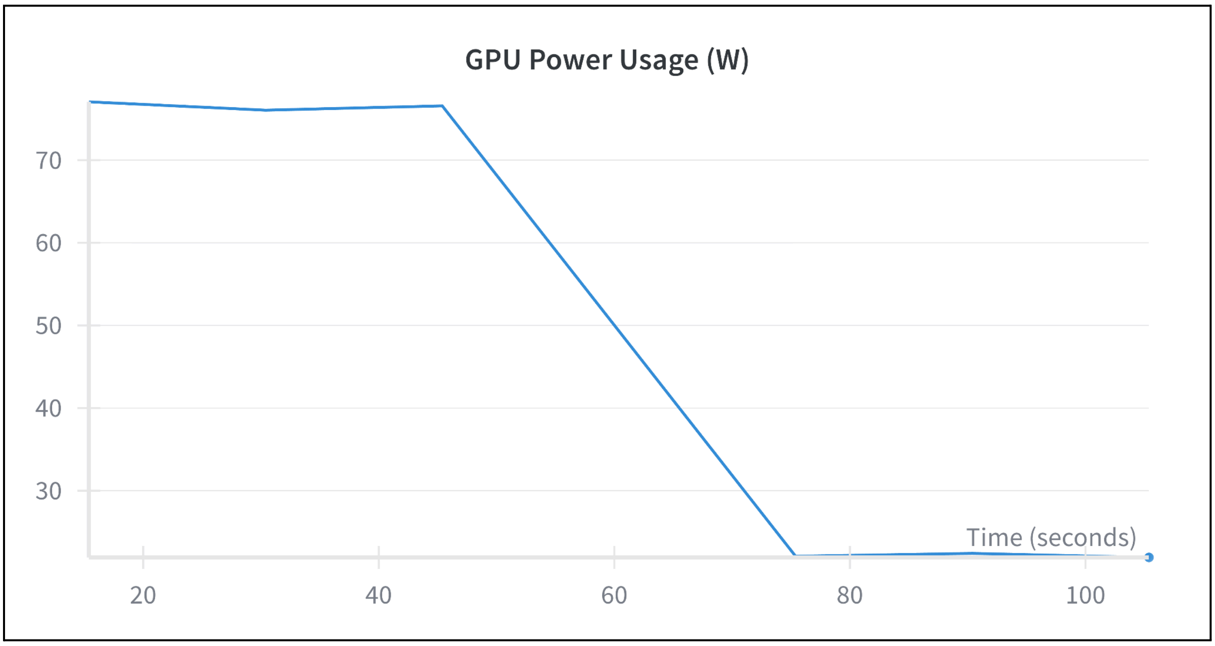
The latest-developed SCAR series of rifles was displayed at the 2013 Latin American Air and Space Defence Fair by FNstar, which consisted mainly of two highly modular rifles of varying calibres and a low-speed grenade launcher of FN40GL40 mm. The SCAR assault rifle was a modular assault rifle designed and developed by the Belgian company FN Hurstar, in cooperation with the United States Special Operations Command, to meet the competitive demand for combat assault rifles for special combat units. The series had two main models, SCAR-L and SCAR-H, of which SCAR-L assault rifles were capable of firing 5.56 x 45 mm NATO ammunition and SCAR-H fired 7.62 x 51 mm NATO ammunition.

Model Response

SCAR

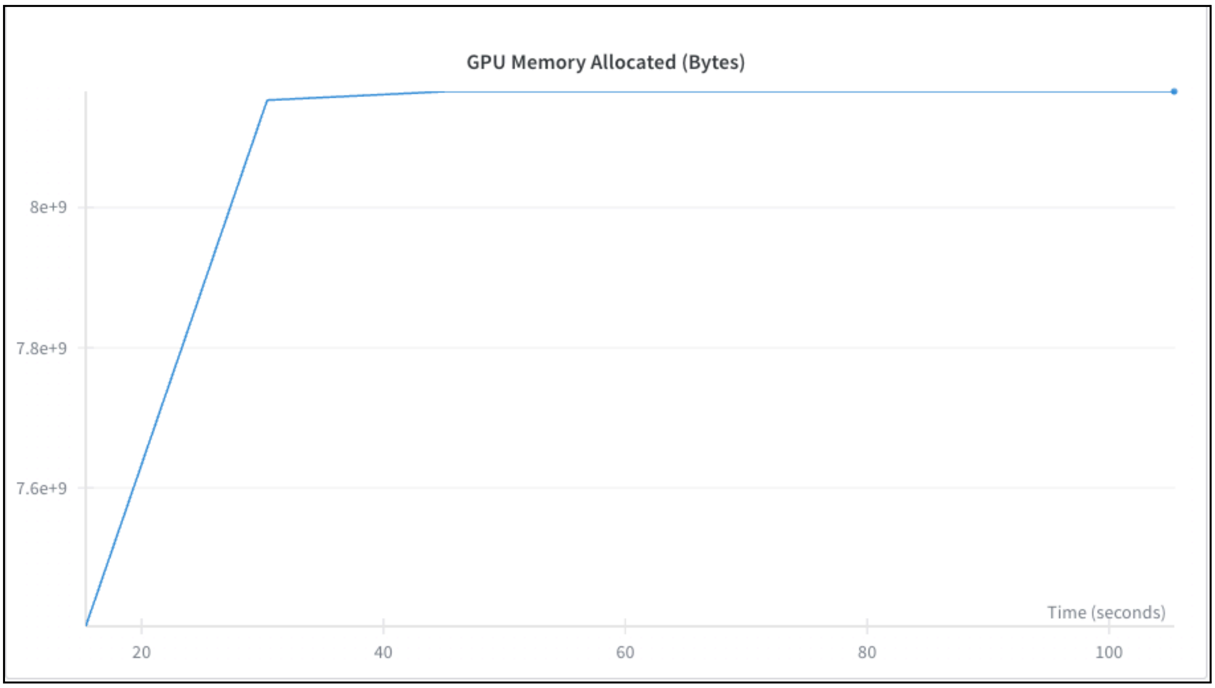
Based on the diagram above, the fine-tuned model successfully answered the question in a simple and precise manner according to the dataset. This shows that the model not only learned the semantic relationships within the data but also improved its ability to generate concise and contextually accurate responses. The improvement reflects better generalization from the training process, where the model can provide meaningful outputs without unnecessary complexity, highlighting the effectiveness of the fine-tuning approach.

GPU Power Usage (W)



The power usage starts high and reaches its peak at 77.055 W before gradually decreasing and stabilizing at 22.069 W. This indicates that the GPU was heavily utilized during the early stages of training, when most computations and parameter updates occurred, and then required less computational power as the training progressed and converged. The stable lower usage at the end suggests that the workload became lighter and more consistent, reflecting efficient resource utilization once the model had settled.

GPU Memory Allocated (Bytes)

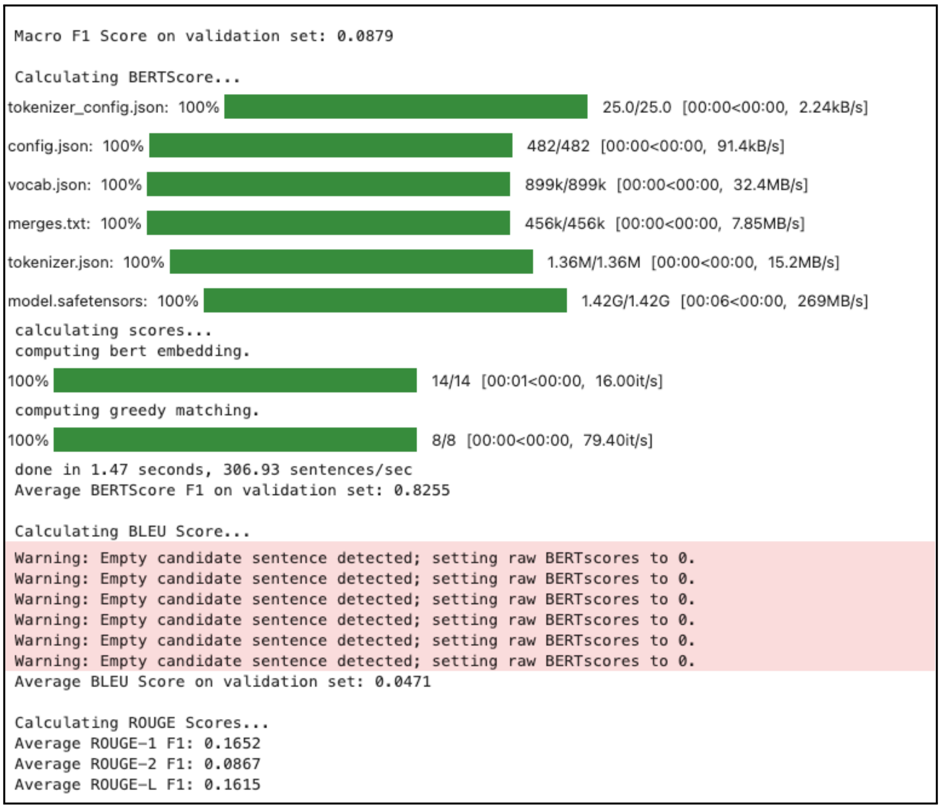


Based on the GPU memory used, the process starts low and gradually increases until it reaches its peak at **8,165,785,600 bytes (8.16 GB)** and then remains constant within that range until the end. This shows that the model progressively loaded parameters and data into memory during the initialization phase, after which memory usage stabilized once all necessary resources were fully allocated. The constant usage indicates efficient memory management, with no additional allocation required throughout the training process.

Top Layer

Source: https://wandb.ai/ahmadhakimiadnan-other/fine-tuning-llms/runs/hzvdyxau/overview

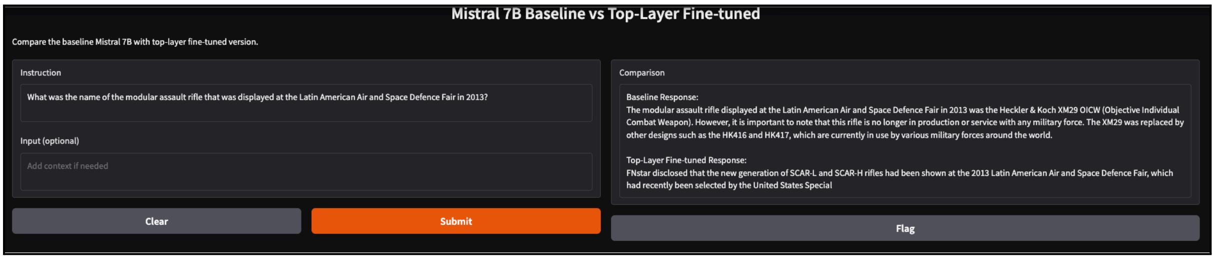
Metrics Evaluation



```
Macro F1 Score on validation set: 0.0879

Calculating BERTScore...
tokenizer_config.json: 100%  25.0/25.0 [00:00<00:00, 2.24kB/s]
config.json: 100%  482/482 [00:00<00:00, 91.4kB/s]
vocab.json: 100%  899k/899k [00:00<00:00, 32.4MB/s]
merges.txt: 100%  456k/456k [00:00<00:00, 7.85MB/s]
tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 15.2MB/s]
model.safetensors: 100%  1.42G/1.42G [00:06<00:00, 269MB/s]
calculating scores...
computing bert embedding.
100%  14/14 [00:01<00:00, 16.00it/s]
computing greedy matching.
100%  8/8 [00:00<00:00, 79.40it/s]
done in 1.47 seconds, 306.93 sentences/sec
Average BERTScore F1 on validation set: 0.8255

Calculating BLEU Score...
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Warning: Empty candidate sentence detected; setting raw BERTscores to 0.
Average BLEU Score on validation set: 0.0471

Calculating ROUGE Scores...
Average ROUGE-1 F1: 0.1652
Average ROUGE-2 F1: 0.0867
Average ROUGE-L F1: 0.1615
```

The evaluation results indicate that the top layer fine-tuned model underperforms compared to previous versions. The Macro F1 score is very low at 0.0879, reflecting poor balance in predicting across different classes and showing difficulty in handling minority categories. Similarly, the BLEU score 0.0471 is also very low. The ROUGE scores ROUGE-1: 0.1652, ROUGE-2: 0.0867, ROUGE-L: 0.1615 remain weak, showing limited lexical overlap with the references. Although the BERTScore F1 0.825 is relatively good, indicating some degree of semantic similarity, the overall results show that the model struggles to produce accurate, consistent, and reliable responses.

Human Evaluation



Mistral 7B Baseline vs Top-Layer Fine-tuned

Compare the baseline Mistral 7B with top-layer fine-tuned version.

Instruction

What was the name of the modular assault rifle that was displayed at the Latin American Air and Space Defence Fair in 2013?

Input (optional)

Add context if needed

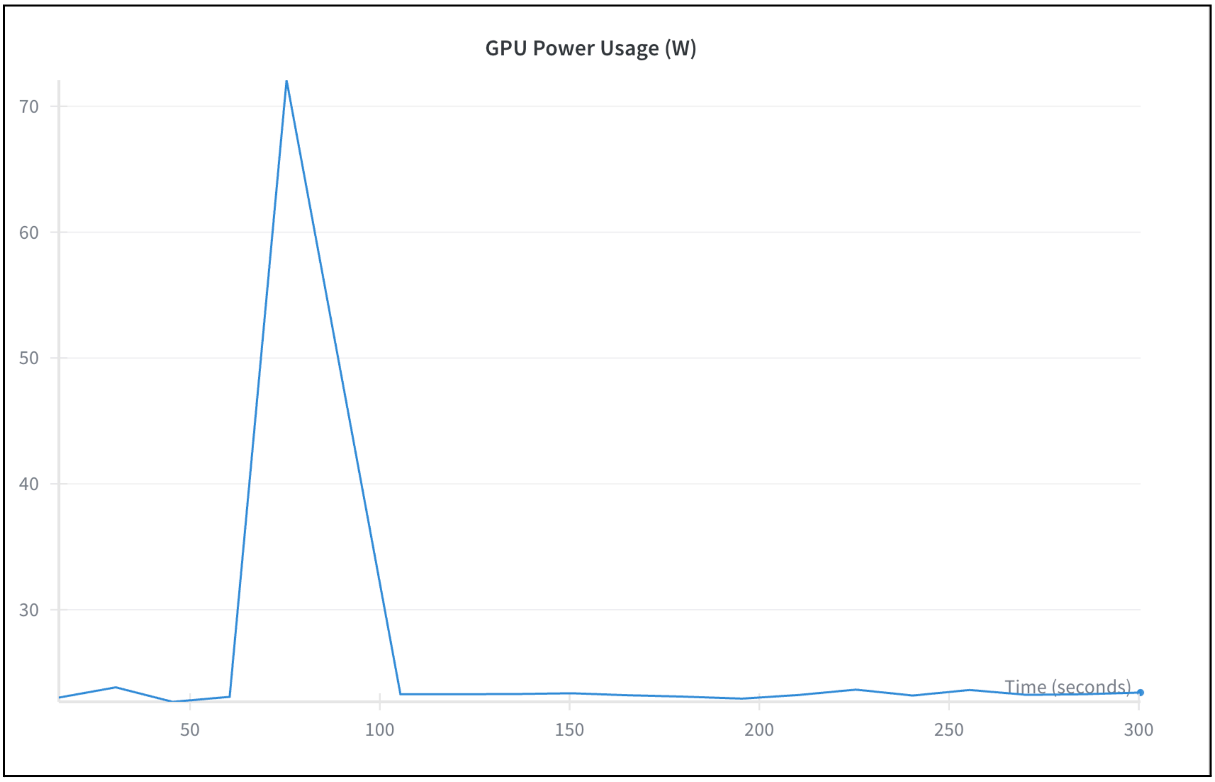
Comparison

Baseline Response:
The modular assault rifle displayed at the Latin American Air and Space Defence Fair in 2013 was the Heckler & Koch XM29 OICW (Objective Individual Combat Weapon). However, it is important to note that this rifle is no longer in production or service with any military force. The XM29 was replaced by other designs such as the HK416 and HK417, which are currently in use by various military forces around the world.

Top-Layer Fine-tuned Response:
FNstar disclosed that the new generation of SCAR-L and SCAR-H rifles had been shown at the 2013 Latin American Air and Space Defence Fair, which had recently been selected by the United States Special

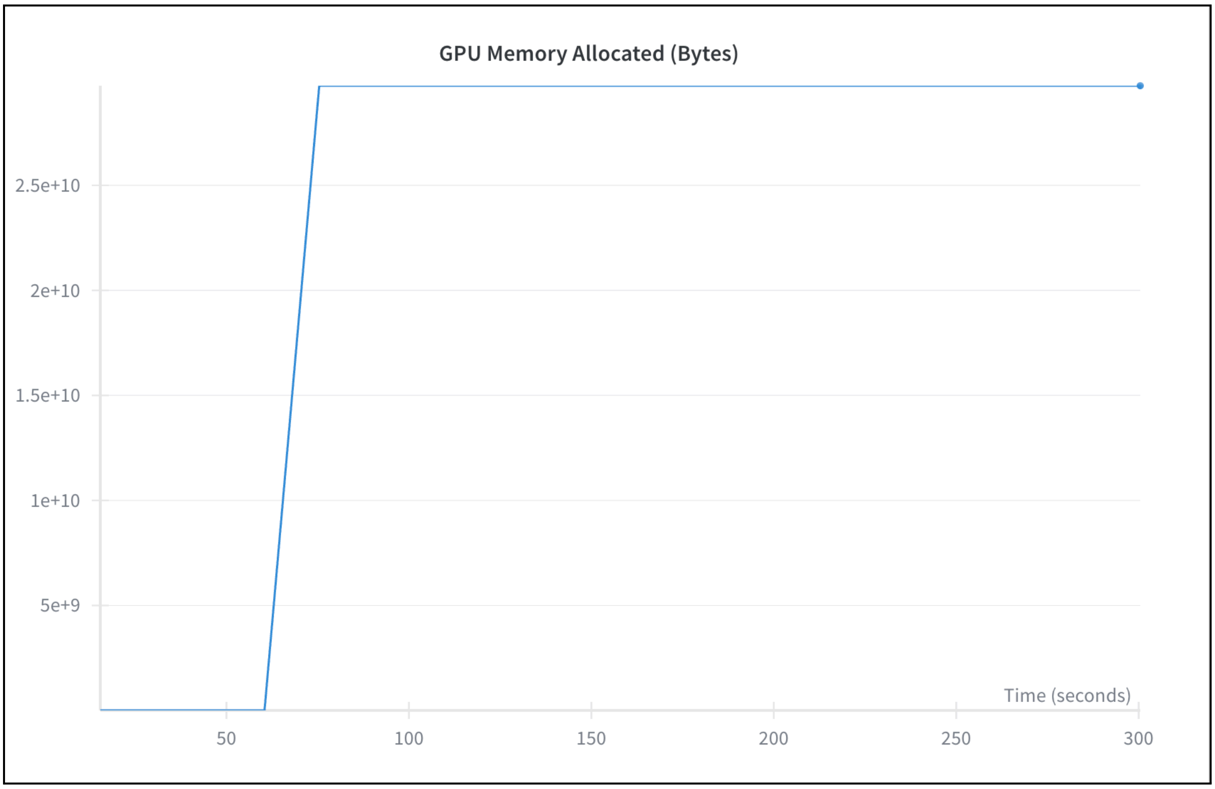
Based on the prompt above, the fine-tuned model gives a correct answer compared to the baseline model. This indicates that the fine-tuning process successfully enhanced the model’s understanding of the dataset, allowing it to generate more accurate and contextually appropriate responses. It shows that even though the overall evaluation metrics are relatively low, the fine-tuned model is capable of learning domain-specific distinctions and producing correct outputs where the baseline model struggled, highlighting the effectiveness of fine-tuning in improving model performance for targeted tasks.

GPU Power Usage (W)



Based on the GPU Power Usage graph, the power consumption begins to increase at around 50 seconds, reaching its peak with 72.075 W, before gradually decreasing and stabilizing at below 30 W until the end. This indicates that the GPU was heavily utilized during the most computationally intensive phase of training, after which the workload reduced and stabilized as the model converged. The steady lower usage toward the end reflects efficient resource utilization once the training process required less computational effort.

GPU Memory Allocated (Bytes)



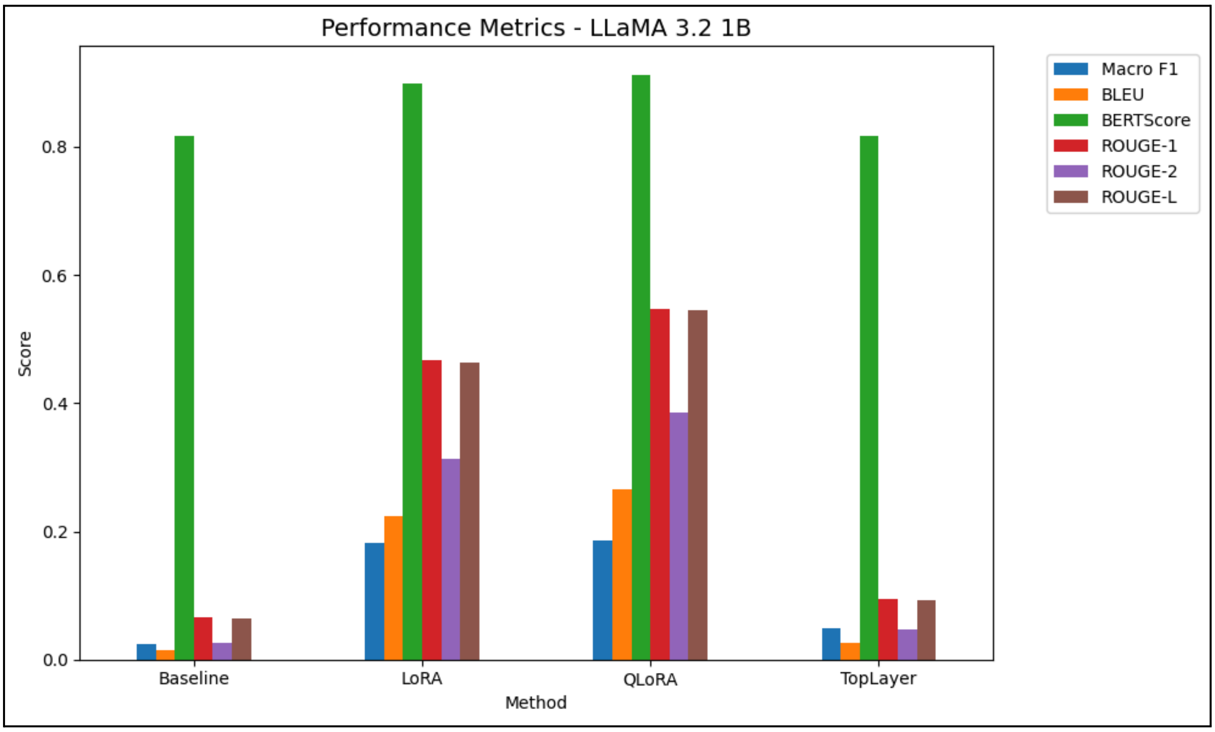
Based on the memory allocation graph, the usage begins to increase at around 60 seconds and reaches a peak of 29,741,744,128 bytes (29.74 GB), after which it remains constant until the end. This indicates that the model gradually loaded all required parameters and data into the GPU memory during the initial phase, and once fully allocated, the memory usage stabilized. The constant allocation throughout the remaining process shows efficient memory management and that no additional memory was required during training.

Table of comparison for Mistral 7B on performance

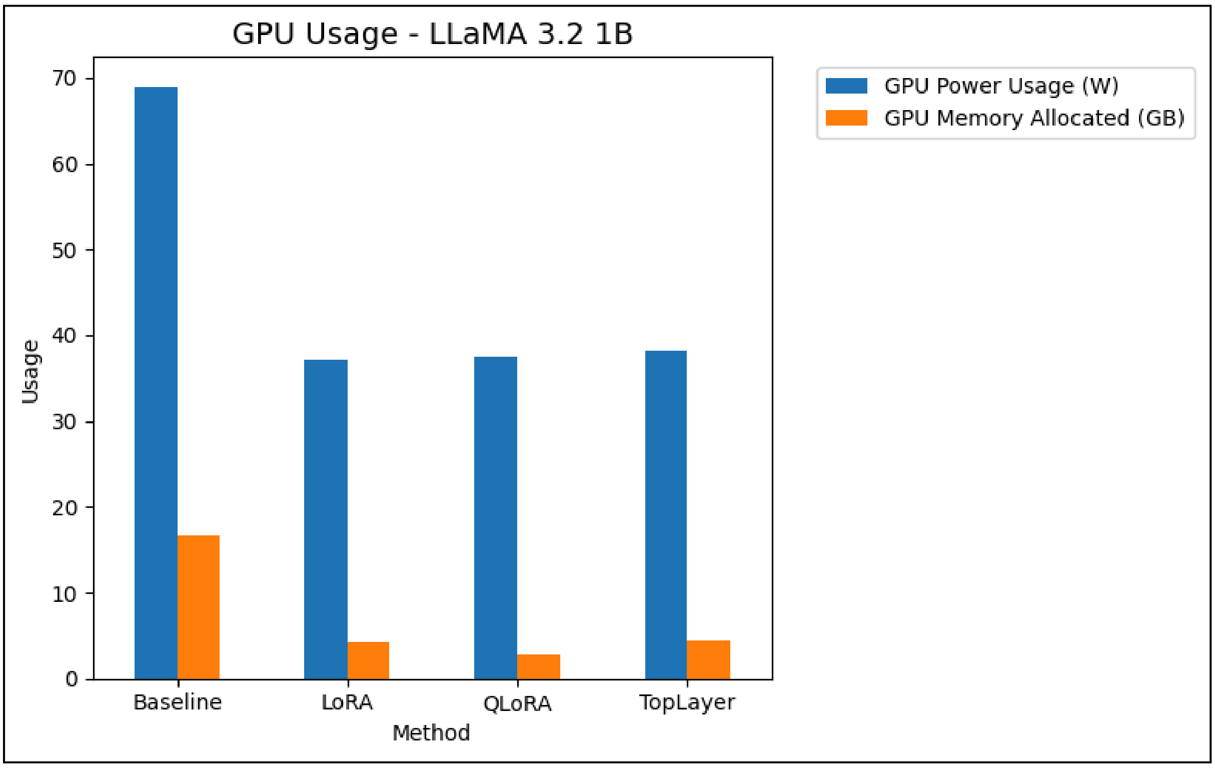
Method	Trainable parameters (%)	Macro F1	BLEU	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L	GPU Power Usage (W)	GPU Memory Allocated (GB)
Baseline	-	0.1041	0.0526	0.8444	0.2072	0.1179	0.2044	48.00	26.15
LoRA	0.0469	0.1998	0.2579	0.9090	0.5250	0.3619	0.5240	72.67	29.75
QLoRA	0.0469	0.5175	0.4700	0.9507	0.7673	0.5497	0.7665	77.06	8.16
TopLayer	7.8700	0.0879	0.0471	0.8255	0.1652	0.0867	0.1615	72.08	29.74

10. Visulization

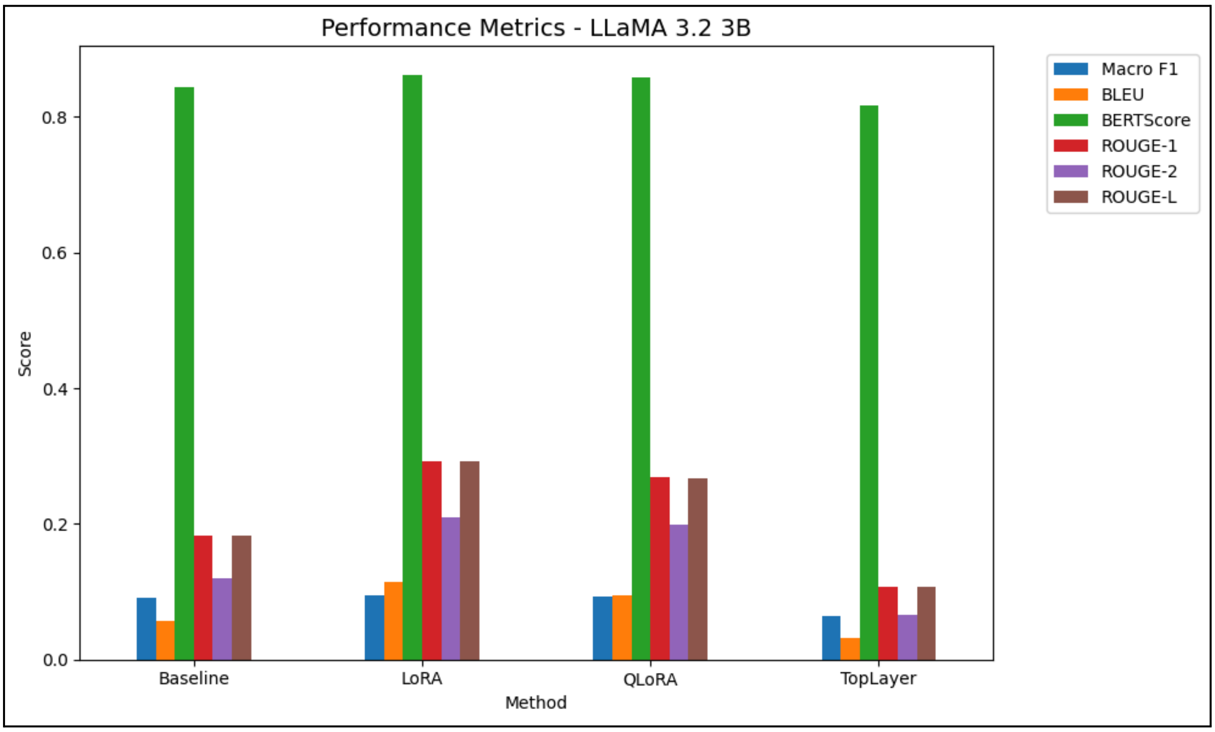
a. Evaluation metrics chart for Llama 3.2 1B across different fine-tuning techniques.



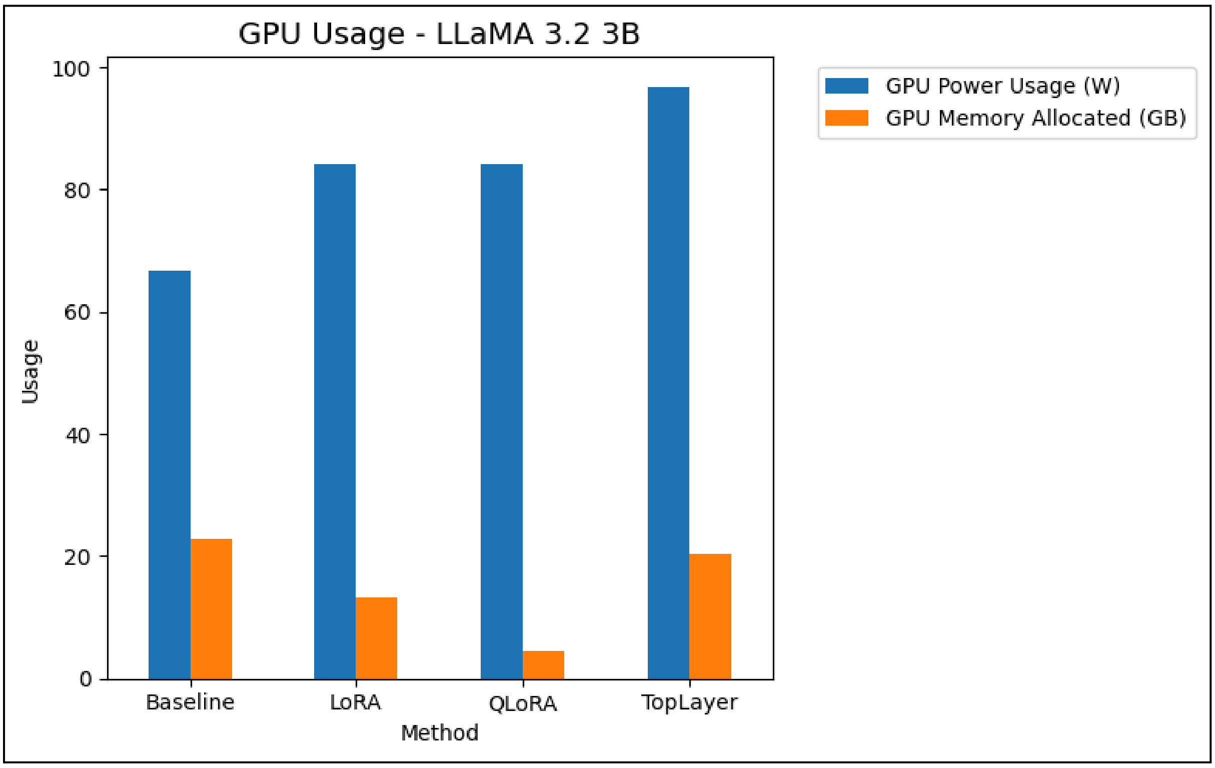
b. Chart of GPU power consumption and memory allocation for Llama 3.2 1B model under different fine-tuning techniques



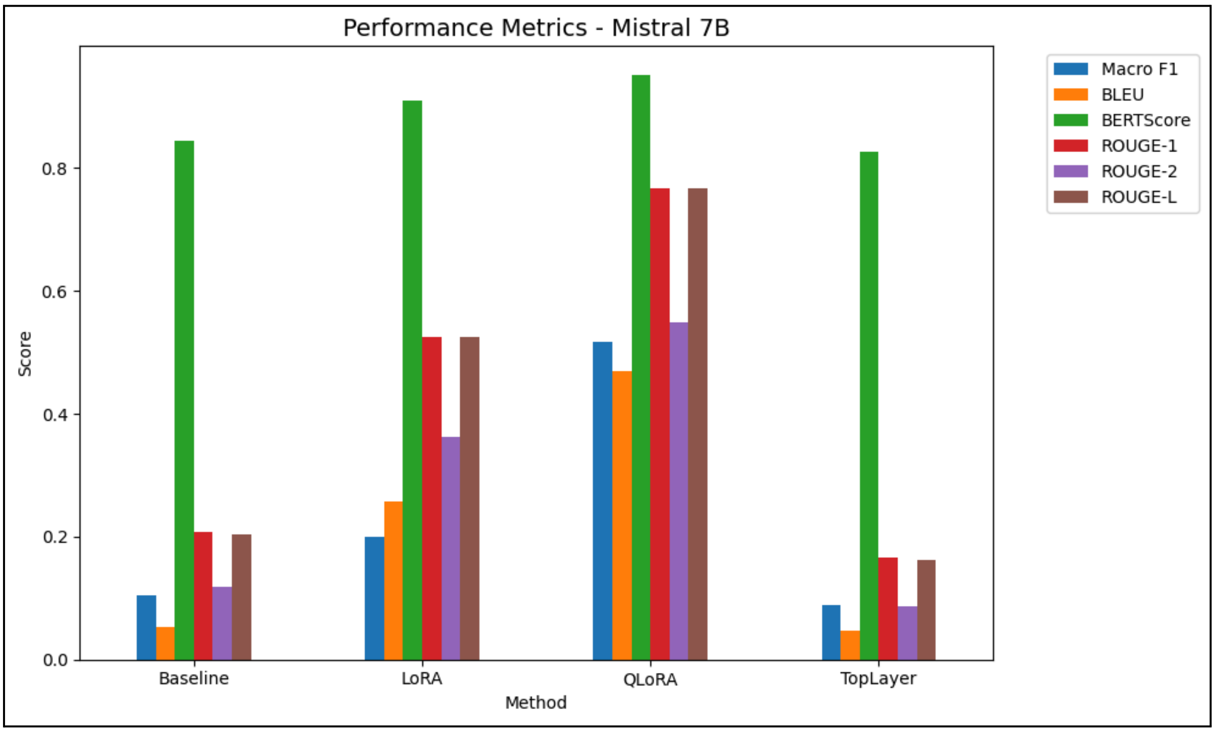
c. Evaluation metrics chart for Llama 3.2 3B across different fine-tuning techniques.



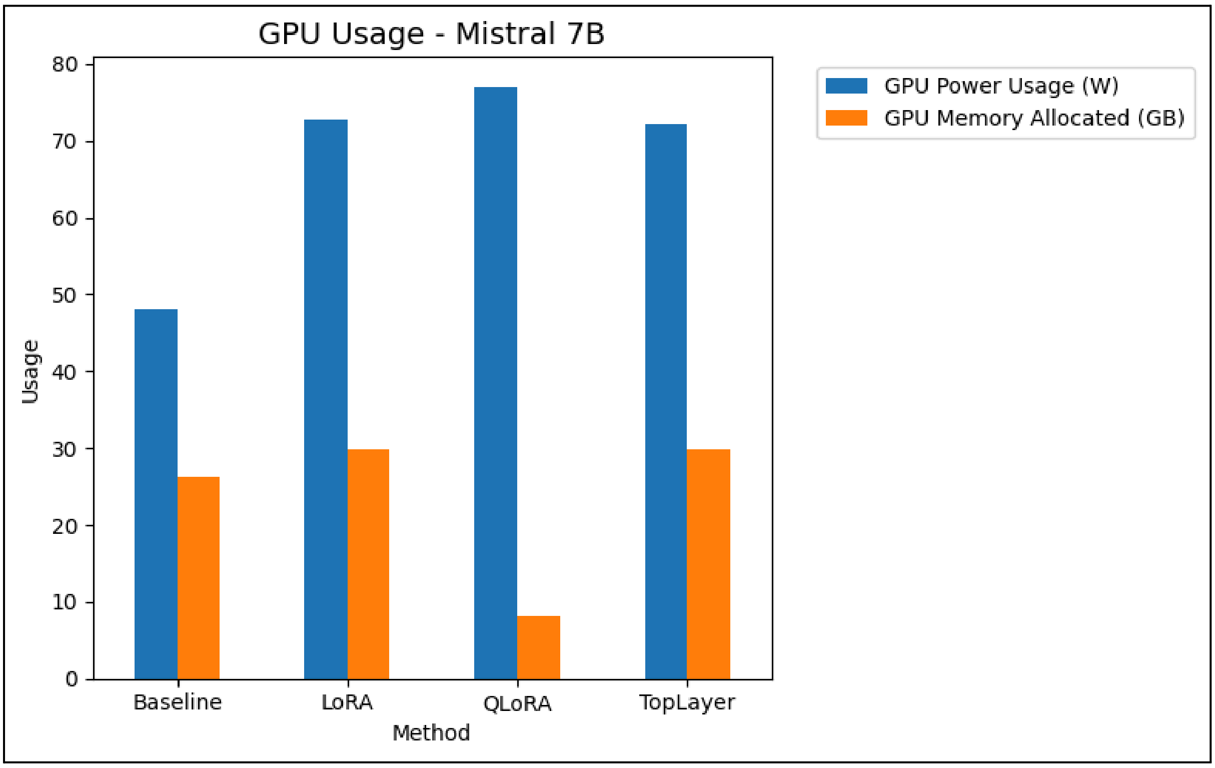
d. Chart of GPU power consumption and memory allocation for Llama 3.2 3B model under different fine-tuning techniques



e. Evaluation metrics chart for Mistral 7B across different fine-tuning techniques.



f. Chart of GPU power consumption and memory allocation for Mistral 7B model under different fine-tuning techniques



11. Conclusion

The optimal method across the evaluated models is QLoRA, as it provides the best balance between performance and resource efficiency. In terms of performance, QLoRA consistently achieves the highest scores across all key metrics, including Macro F1, BLEU, BERTScore, and ROUGE, particularly excelling in larger models such as Mistral 7B. Despite this strong performance, QLoRA trains only a tiny fraction of parameters (~0.0469–0.069%), significantly reducing memory usage (2.74 GB for LLaMA 1B, 8.16 GB for Mistral 7B) and maintaining moderate GPU power consumption (37–77 W). This combination of high performance and minimal resource demand makes QLoRA the most efficient and practical method for fine-tuning large language models.

12. References

1. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685.

2. *Stanford CRFM*. (2023, March 13). https://crfm.stanford.edu/2023/03/13/alpaca.html

3. *sentence-transformers/all-MiniLM-L6-v2 · Hugging Face*. (2024, January 5). https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

4. Chawla, A. (2024, March 1). Full-model Fine-tuning vs. LoRA vs. RAG. *Daily Dose of Data Science*. https://blog.dailydoseofds.com/p/full-model-fine-tuning-vs-lora-vs

5. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... & Scao, T. L. (2023). *LLaMA: Open and Efficient Foundation Language Models*. arXiv preprint arXiv:2302.13971.

6. Unsloth Documentation. (2024). https://github.com/unslothai/unsloth

7. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of ACL.

8. *BLEU - a Hugging Face Space by evaluate-metric*. (n.d.). https://huggingface.co/spaces/evaluate-metric/bleu

9. *IBM watsonx as a Service*. (n.d.). https://www.ibm.com/docs/en/watsonx/saas?topic=metrics-exact-match

10. Kundu, R. (n.d.). F1 Score in Machine Learning: Intro & Calculation. *V7*. https://www.v7labs.com/blog/f1-score-guide

11. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019, April 21). *BERTScore: Evaluating Text Generation with BERT*. arXiv.org. https://arxiv.org/abs/1904.09675

12. *SAG: Style-Aligned article Generation via model collaboration*. (n.d.). https://arxiv.org/html/2410.03137v1#S4

13. *LLM Evaluation Metrics: The Ultimate LLM Evaluation Guide - Confident AI*. (n.d.). https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation

14. Jain, R. (2023, August 22). LLMS — Fine-tuning and model Evaluation - Ritik Jain - medium. *Medium*. https://ritikjain51.medium.com/llms-fine-tuning-and-evaluation-f019515b1c67

15. Fitzmaurice, G. (2024, April 19). Meta's Llama 3 will force OpenAI and other AI giants to up their game. *IT Pro*.

https://www.itpro.com/technology/artificial-intelligence/metas-llama-3-will-force-openai-and-other-ai-giants-to-up-their-game?

16. Stats, L. (n.d.). *Gemma 2 9B vs Qwen2.5 7B Instruct*. LLM Stats. https://llm-stats.com/models/compare/gemma-2-9b-it-vs-qwen-2.5-7b-instruct

17. Science, O.-. O. D. (2025, March 13). The Best Lightweight LLMs of 2025: Efficiency Meets Performance. *Medium*. https://odsc.medium.com/the-best-lightweight-llms-of-2025-efficiency-meets-performance-78534ce45ccc

18. Aswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017).
Attention is all you need. In *Advances in Neural Information Processing Systems* (NeurIPS), 30.
https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

19. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).
BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171–4186.
https://doi.org/10.18653/v1/N19-1423

20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019).
Language models are unsupervised multitask learners. OpenAI.
https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

21. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Jegou, H. (2023).
LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
https://arxiv.org/abs/2302.13971

22. Howard, J., & Ruder, S. (2018).
Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328–339).
https://doi.org/10.18653/v1/P18-1031

23. Pan, S. J., & Yang, Q. (2010).
A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359.
https://doi.org/10.1109/TKDE.2009.191