Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/21048
DOI URL: http://dx.doi.org/10.21474/IJAR01/21048

RESEARCH ARTICLE

QUESTION-ANSWER SYSTEM ON MEDICAL DOMAIN WITH LLMS USING VARIOUS FINE-TUNING METHODS

Misha Patel, Mansi Kotadiya and Urvashi Solanki
1.P.P Savani University, Kosmba, Surat, Gujarat, India.

Manuscript Info

Manuscript History
Received: 27 March 2025
Final Accepted: 30 April 2025
Published:May 2025

Key words:-
Medical, LLMS, Finetuning

Abstract

The challenge of developing artificial intelligence (AI) with the ability to comprehendand produce human language has persisted since the 1950s, when the TuringTest was first proposed. Language modelling techniques have advanced fromstatistical to neural models, recently focusing on pre-trained language models(PLMs) utilizing Transformer architecture. These PLMs, trained on vast datasets,excel in various natural language processing (NLP) tasks. Researchers have discoveredthat increasing the size of these models enhances their capabilities andeven imparts unique abilities like in-context learning and the ability to think likehuman brains. These more significant variants are referred to as large languagemodels (LLMs). This report examines recent LLM advances, encompassing pretraining,adaptation tuning, utilization, and capacity evaluation on specificallymedical domains with not-so-large language models. Also, work with the PEFTLibraries like the LoRa and QLora techniques to accommodate LLMs on a singleGPU. Index Terms—Pre-trained language models(PLMs), ChatGPT, Large languagemodels(LLMs), Finetuning, Promt engineering, Reinforcement learning withhuman feedback, Chain-Of-Thoughts.

Introduction:-

Artificial Intelligence (AI) has achieved remarkable progress in areas like natural language processing, image recognition, and decision-making. However, its application in medicine remains limited due to challenges related to trust, interpretability, and alignment with human expertise. Diagnostic accuracy is a persistent issue in the medical field, where even experienced clinicians occasionally misidentify conditions due to symptom complexity or data limitations. Our research investigates how large language models (LLMs), when fine-tuned with domain-specific medical data and connected to external knowledge sources, can improve diagnostic support. These models can offer context-aware, accurate suggestions by analyzing patient records at scale. This forms the foundation for a new form of human-AI collaboration, where AI systems learn continuously from human feedback but operate autonomously for lower-level tasks. In this paradigm, human interaction is limited to high-level guidance, correction, and critique.

Building upon prior work in human-aligned AI and reward modeling, our approach focuses on reducing training costs and model complexity by employing efficient fine-tuning strategies. We utilize open-source LLMs tailored for

Corresponding Author:- Misha Patel
Address:- P.P Savani University, Kosmba, Surat, Gujarat, India.

specific diseases or medical environments to ensure compatibility with lower-resource systems such as standard CPUs. Key technologies in our pipeline include pre-trained APIs from Google, Meta's ASR models, and various open-source LLMs like GPT-3, BERT, T5, RoBERTa, BLOOM, Falcon, Dolly, LLaMA, and Mistral [1][2]. To further enhance medical relevance, we integrate Retrieval-Augmented Generation (RAG) models [3] for external data access and Chain-of-Thought prompting [4] to improve logical reasoning in responses. Our application aids clinicians by answering patient questions and recommending treatments, blending reinforcement learning with supervised learning techniques. This research introduces a low-cost, scalable, and domain-adaptable AI approach tailored to medical diagnostics. The following sections elaborate on the system architecture, model optimization techniques, and performance assessment.

State-of-the-Art

Large Language Models (LLMs) have been the subject of a great deal more research in recent years, mostly because of their revolutionary potential in a variety of application domains. These models have shown significantusefulness in fields including healthcare [5], banking [6], education [7], and law [8], where they carry out duties like document classification, sentiment analysis, text summarizing, and question answering. Understanding the fundamental architecture and operational needs of LLMs is crucial given the increased interest in implementing them on contexts with limited resources, including CPU-based systems or edge devices. To make LLMs appropriate for these platforms, methods including knowledge distillation, model quantization, and pruning are being investigated [9], [10]. Therefore, this section begins with a foundational overview of LLMs, including their structure, cross-domain performance, and strategies for efficient deployment on low-power devices.

Background for Large Language Models (LLMs)

The development of artificial intelligence (AI), especially in the area of natural language processing (NLP), has relied heavily on large language models (LLMs). Large amounts of text are used to train these models, which are based on the transformer architecture [11].corpora and have proven their capacity to produce logical, human-like language, comprehend context, and complete a range of natural language processing (NLP) tasks, including question answering, translation, and summarization. In order to enable a broad variety of generalization skills across domains, LLMs learn the statistical correlations between words, sentences, and contexts [12].

Examples of Large Language Models

Several popular and important LLMs for research are as follows:

Generative Pre-trained Transformer 3, or GPT-3: GPT-3, an autoregressive language model created by OpenAI, hasKnown for its few-shot and zero-shot learning capabilities, 175 billion parameters [13].Transformer-Based Bidirectional Encoder Representations, or BERT: BERT, which was first introduced by Google, achieves state-of-the-art performance in numerous NLP tasks by using a masked language model and next sentence prediction to grasp context in both directions [14]. XLNet: Developed by Google Brain and Carnegie Mellon University researchers, XLNet combines concepts from permutation-based language modeling and auto-regressive models, surpassing BERT on a number of benchmarks [15].Google created T5 (Text-to-Text Transfer Transformer), which unifies various task formats into a single model architecture by treating each NLP task as a text-to-text transformation problem [16].Facebook AI Research introduced RoBERTa (Robustly Optimized BERT Pretraining Approach), a variation of BERT that improves performance by using larger batches of training data and eliminating the next sentence prediction aim [17].

Figure 2.1 illustrates these popular LLMs, summarizing their architecture, training methods, and key contributions.

Examples of Open-Source LLMs

While many large language models (LLMs) are proprietary and not freely accessible for commercial applications, the emergence of open-source LLMs has significantly advanced the natural language processing (NLP) landscape. These models provide developers, researchers, and organizations with valuable tools to experiment, innovate, and deploy NLP-driven solutions. Open-source LLMs lower the barrier to entry by enabling wider access to powerful language modeling capabilities, thus supporting both academic exploration and commercial product development.

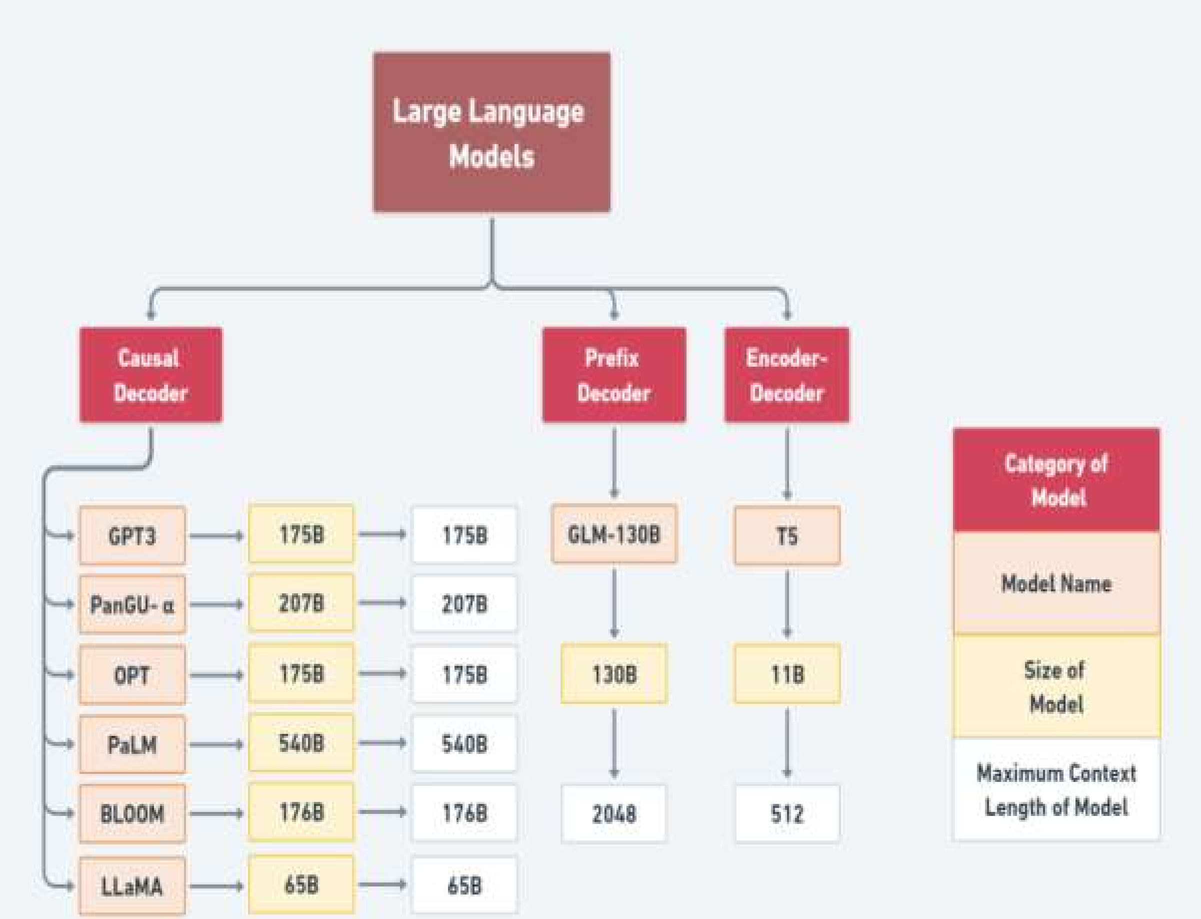


Figure 2.1:- Representative Examples of Popular Large Language Models (LLMs).

A number of open-source large language models (LLMs) have been developed to promote transparency, accessibility, and research innovation in natural language processing. **BLOOM** (BigScience Large Open-science Open-access Multilingual Language Model), developed by the BigScience research collaboration, is designed for multilingual tasks and openly released for research and commercial use under a responsible licensing framework [18]. **Falcon**, created by the Technology Innovation Institute (TII), is another high-performing open-source model optimized for efficiency and scalability in real-world applications [19]. **LLaMA 2**, released by Meta (formerly Facebook), has been fine-tuned using Reinforcement Learning from Human Feedback (RLHF) to enhance safety and performance in dialogue and general NLP tasks [20]. **Guanaco**, developed by the UW NLP group, incorporates the Low-Rank Adaptation (LoRA) fine-tuning technique, introduced by Tim Dettmers et al., enabling efficient adaptation of LLMs on limited computational resources [21]. Additionally, **GPT-NeoX-20B**, an autoregressive transformer model developed by EleutherAI, demonstrates competitive performance with proprietary models and serves as a foundation for open research and experimentation in scalable LLMs [22].

Examples of Large Language Models Specialized in the Medical Domain

Med-PaLM, created by Google Research, is one of the noteworthy big language models specifically designed for the medical field.The MultiMedQA dataset, which is especially selected for medical question-answering tasks, has been used to refine Med-PaLM.Figure 2.2 illustrates the datasets used to train the PaLM model in the medical domain, highlighting the specialized data sources that enhance its performance on healthcare-related applications.

LLM: BLOOM Model

The BLOOM model has been developed in multiple versions through the BigScience Workshop, an initiative inspired by collaborative open science projects where researchers pool resources and expertise to maximize collective impact [23]. Architecturally, BLOOM is based on an autoregressive transformer similar to GPT-3, designed for next-token prediction. However, BLOOM distinguishes itself by being trained on a multilingual corpus comprising 46 natural languages and 13 programming languages. Various smaller versions of BLOOM have also been trained on this dataset, including bloom-560m, bloom-1b1, bloom-1b7, bloom-3b, bloom-7b1, and the full-scale bloom-176b with 176 billion parameters.

The BLOOM transformer includes a span classification head, enabling extractive question-answering tasks such as those exemplified by the SQuAD dataset. This classification head is implemented as a linear layer atop the hidden states output to compute logits for span start and end positions.

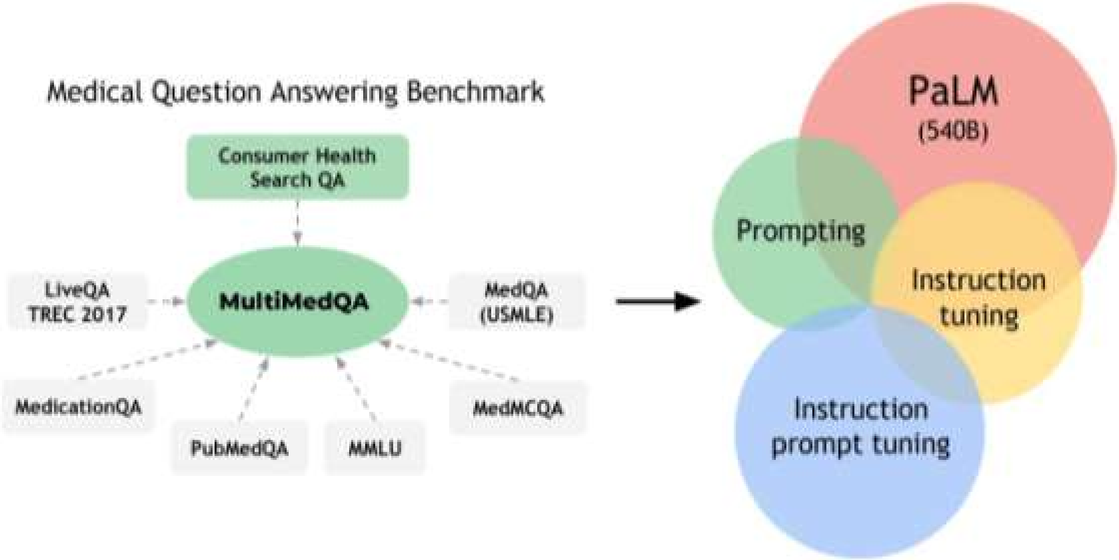


Figure 2.2:- A large language model (LLM) called Med-PaLM was created to offer superior responses to medical queries.

After fine-tuning the BLOOM version-2 3-billion parameter model using QLoRA—a parameter-efficient fine-tuning technique—the updated model configuration is illustrated in Figure 2.3.

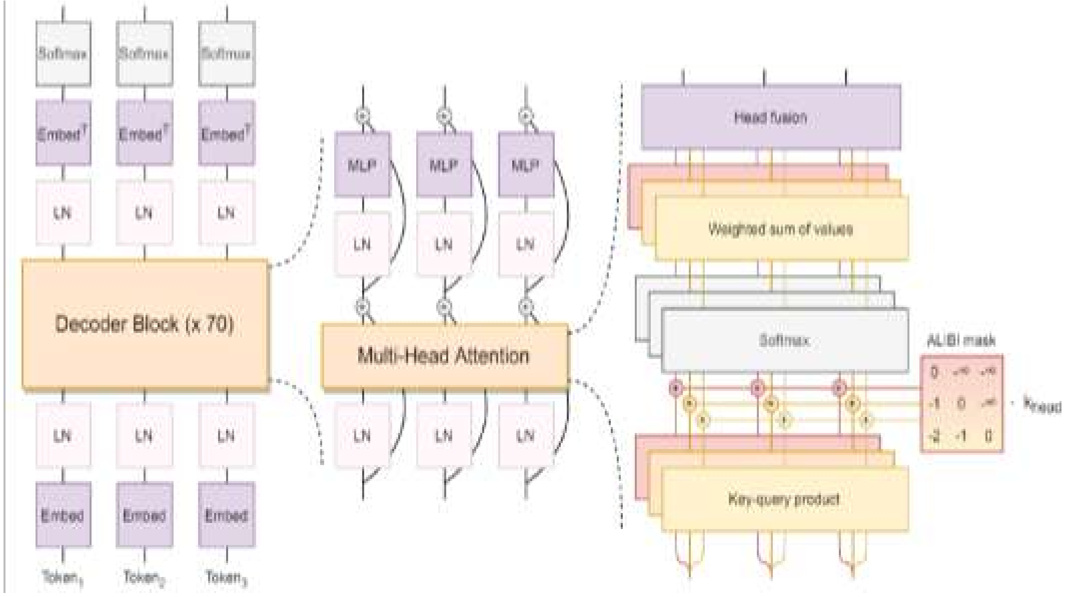


Figure 2.3:- The BLOOM Architecture [22].

Proposed Approach

Figure 3.1 shows how a voice-based QA system for a particular domain works with LLM. It takes voice input and processes it to text, then apply to LLM and gets answers from it, then gets better results using the Reinforcement learning model with the human feedback model, and finally gets output answers in the form of the audio file.

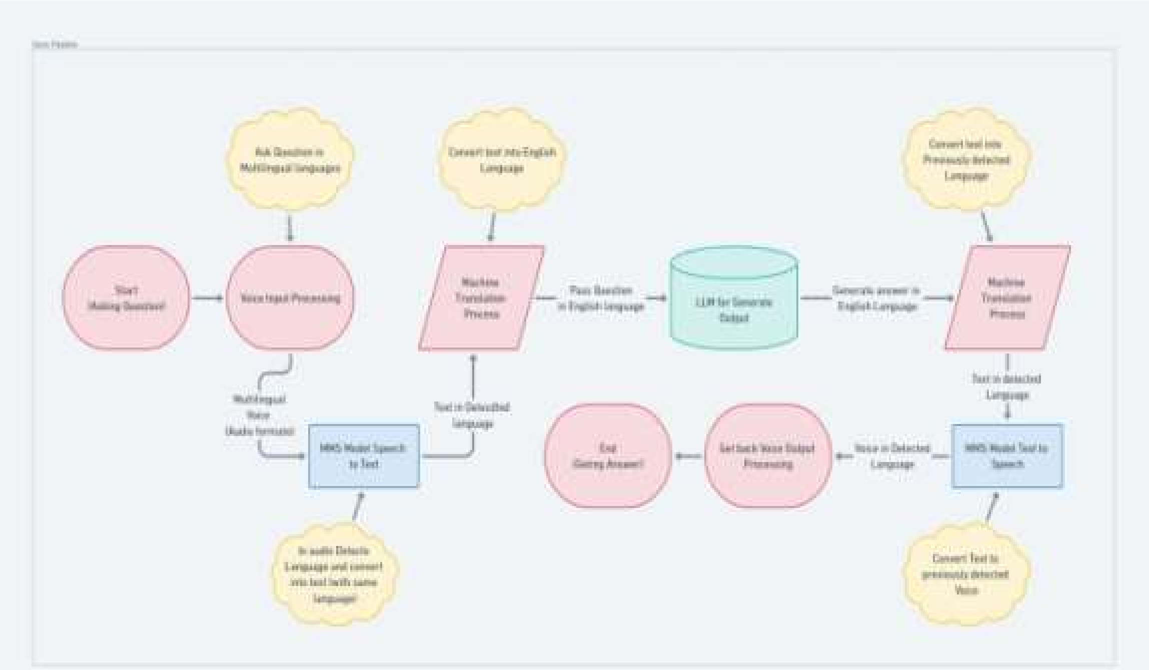

Ask Question in Multilingual languages
Convert text into English Language
Convert text into Previously detected Language
Start (Asking Question)
Voice Input Processing
Machine Translation Process
Pass Question in English language
LLM for Generate Output
Generate answer in English Language
Machine Translation Process
Multilingual Voice (Audio formate)
MMS Model Speech to Text
Text in Detected language
Text in detected Language
End (Getting Answer)
Get back Voice Output Processing
Voice in Detected Language
MMS Model Text to Speech
In audio Detects Language and convert into text (with same language)
Convert Text to previously detected Voice

Figure 3.1:- Basic pipeline for QA system using LLM.

We mainly divided the whole pipeline into three phases. The first phase is before the LLM part, the second phase is theLLM work, and the last phase is after getting results from the LLM, which is further explained in depth.

Phase I: Speech-to-Text & Translation Part

The interaction unfolds with the user initiating the process by posing a question, triggering the activation of the voice module designed for input processing. This module transforms the user's spoken words into text, creating a foundation for further analysis. We try Facebook wav2vec [24] and Openai whisper [25] ASR model API to achieve this. The system incorporates a translation feature for questions in low-resource languages such as Hindi, Gujarati, Bengali, Tamil, and others to ensure inclusivity and accommodate diverse linguistic preferences. This multilingual capability broadens the system's reach, facilitating seamless communication across language barriers. The translated question in English then undergoes the next phase, where it is fed into a specialized Language Model (LLM) system.

Phase II: Finetuning LLMs

Unlike larger and more generalized language models like ChatGPT, the uniqueness of this system lies in its utilization of low-parameter models specifically curated for domain-specific question answering. Despite their reduced complexity, these models are adept at comprehending and responding to queries with accuracy and relevance comparable to their larger counterparts.The LLM processes the input question, utilizing its domain-specific knowledge to generate a coherent and contextually appropriate response. This ensures the information provided is accurate and tailored to the domain under consideration. [26,27,28] Using low-parameter models balances computational efficiency and generates meaningful responses, making the system well-suited for targeted applications. To achieve this, we use PEFT(Parameter Efficient Finetuning) libraries like LoRa and QLoRa techniques specifically for reducing the parameters and other prompt engineering with RAG, RLHF, and Chain-of-Thoughts finetuning techniques to make the response more relatable and accurate.

Phase III: Back Traslation & Text-to-Speech Part

Upon receiving the response in English text from the LLM, the final step involves translating the answer back to the user's original language. This translation is then transformed into audio format, employing a comprehensive approach to deliver the system's output. For that, we again use the same Google API for back translation and then use the MMS-TTS(Massively Multilingual Speech project & Text-to-Speech) model to get the audio answer in our specific language. This entire process, orchestrated by low-parameter language models, exemplifies an effective and specialized method for voice-based question answering. By accommodating various languages and leveraging domain-specific expertise, the system ensures that users receive accurate and contextually relevant information, enhancing accessibility and user experience.

Dataset

We used 20K questions for our training part, which we made from the two different datasets mentioned below. The data statistics are given below,

Table 3.1:- Data Statistics used for.

	MedMCQA	USMLE from MedQA
Train	11,218	8,790
Test	100	100

MedMCQA

In order to handle actual medical entrance exam questions, MedMCQA is a large-scale multisubject multichoice dataset for medical domain question answering.

With an average token length of 12.77 and a high thematic diversity, MedMCQA offers approximately 194k excellent multiple-choice questions (MCQs) for the AIIMS and NEET PG entrance exams that cover 2.4k healthcare themes and 21 medical subjects. An open-source dataset for the field of natural language processing is offered by MedMCQA. It is anticipated that this dataset will aid future studies aimed at improving QA systems. Data statistics are displayed in Table 3.2.

Table 3.2:- Data Statistics Of MedMCQA.

	Train	Test	val
Questions #	182,822	4,183	6,150
Vocab	94,231	11,218	10,800
Max Ques. Tokens	220	135	88
Max Ans. Tokens	38	21	25

Data Instances

```
{
    " question " : "A 40−year−old man presents with five days of producti cough and fever . Pseudomonas aeruginosa is isolated from a pulmona abscess . CBC shows an acute effect characterized by marked leukocy (50 ,000 mL) , and the differential count reveals a shift to the lef hematologic findings ?"
    " exp " : " Circulating levels of leukocytes and their precursors may occasionallyreach very high levels ( >50 ,000 WBC mL) . These extreme are similar to the white cell counts observed in leukaemia , which rise in the number of mature and immature neutrophils in the blood , referred to as a shift to the left . In contrast to bacteria decrease in the circulating WBC count . "
    " cop " : 1 ,
     "opa " : "Leukemoid reaction " ,
     "opb " : " Leukopenia " ,
     " opc " : " Myeloid metaplasia " ,
     "opd " : " Neutrophilia " ,
       " subject_name " : " Pathology " ,
      " topic_name " : " Basic Concepts and Vascular Changes of Acute Inflammation " ,
    " id " : " 4 e1715fe −0bc3 −494e−b6eb−2d4617245aef " ,
       " choice_type " : " Single "
}
```

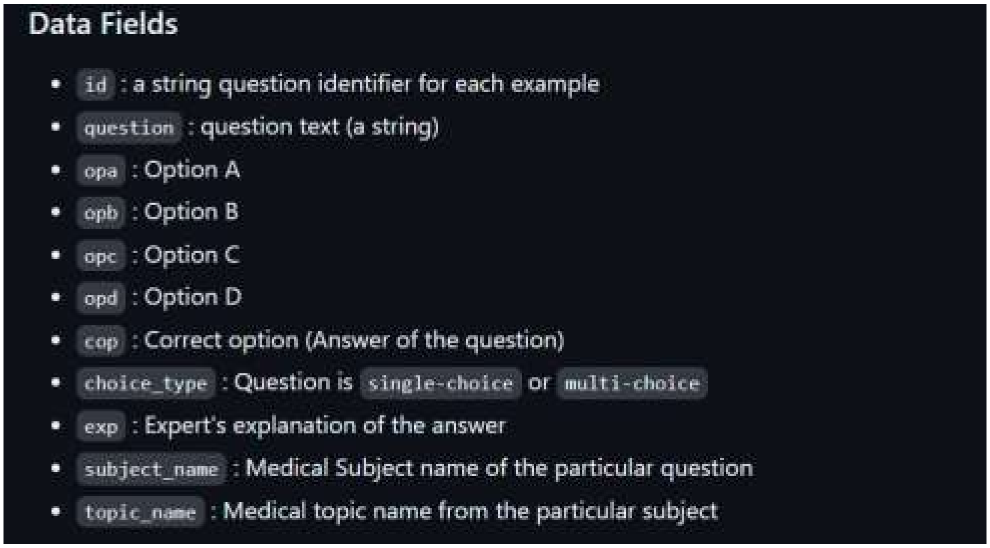
Data Fields

Figure 3.2 shows the question or record's different fields.

USMLE from MedQA

We tackle medical challenges and simulate a challenging real-world scenario using MEDQA, a new OpenQA dataset.

This dataset's questions are taken from US medical board exams, which assess medical professionals' professional expertise and clinical judgment [29]. We only use questions from the National Medical Board Examination in the USA, however there are also questions from medical board exams in Taiwan and mainland China. Table 3.3 presents their data statistics.



Data Fields

- `id` : a string question identifier for each example
- `question` : question text (a string)
- `opa` : Option A
- `opb` : Option B
- `opc` : Option C
- `opd` : Option D
- `cop` : Correct option (Answer of the question)
- `choice_type` : Question is `single-choice` or `multi-choice`
- `exp` : Expert's explanation of the answer
- `subject_name` : Medical Subject name of the particular question
- `topic_name` : Medical topic name from the particular subject

Figure 3.2:- Data Formate of MedMCQA dataset [29].

Table 3.3:- Data Statistics of USMLE.

Metric	USMLE
# of options per question	4
Avg./Max. Option len.	3.5 / 45
Avg./Max. Question len.	116.6 / 530
vocab/character size	63317
# of questions in Train	10178
# of questions in Development	1272
# of questions in Test	1273

Data Instances

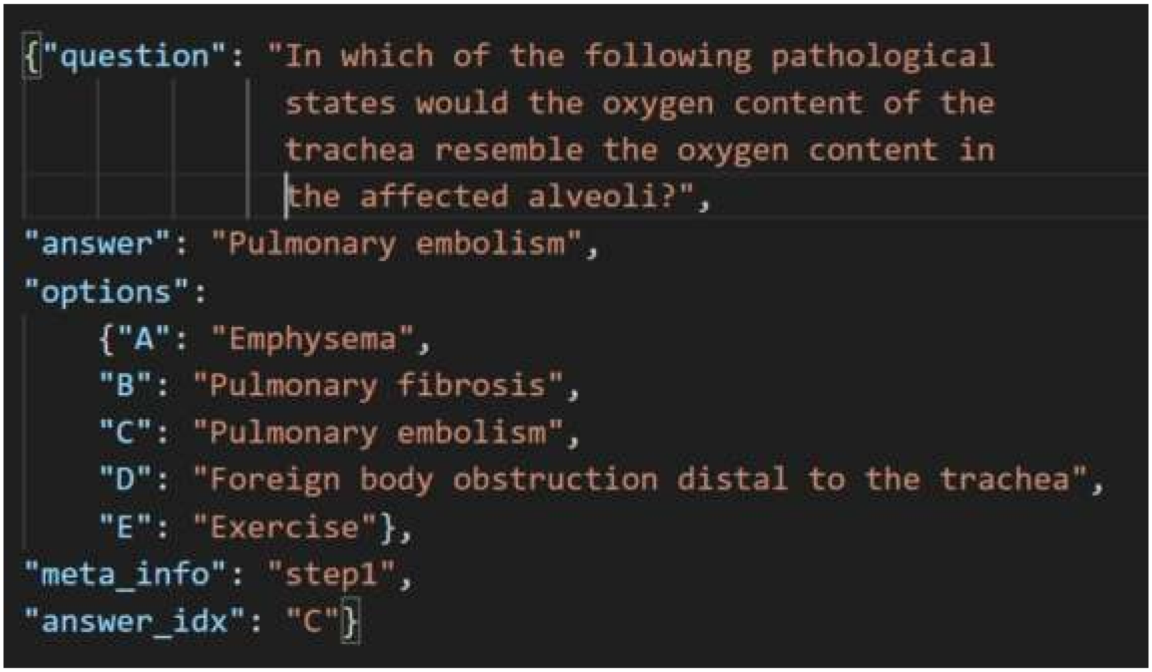
There are two types of questions in USMLE data: 1) The question asks for the patient's symptoms; 2) it analyzes the patient's condition first, then asks for the most likely diagnosis, course of treatment, necessary examination, etc. Figure 3.3 displays the data record's comprehensive information.

Techniques for Finetuning LLMs

Overview

Finetuning existing LLMs improves the model performance for the domain-specific use case for our project, which is the medical domain. We can show that the fine- tuning LLMS is quite similar to supervised learning methods. Here are some steps to perform instruction finetuning: preparing training data, dividing it into splits, passing prompts to the model, comparing it with desired responses, calculating loss, and updating model weights. And their Outcome: An improved version of the base model known as an instruct model. Figure 3.4 shows the difference between base and fin-tuned model output.

Following are some Adaptation Tuning of LLMs,



```
{"question": "In which of the following pathological
              states would the oxygen content of the
              trachea resemble the oxygen content in
              the affected alveoli?",
"answer": "Pulmonary embolism",
"options":
    {"A": "Emphysema",
    "B": "Pulmonary fibrosis",
    "C": "Pulmonary embolism",
    "D": "Foreign body obstruction distal to the trachea",
    "E": "Exercise"},
"meta_info": "step1",
"answer_idx": "C"}
```

Figure 3.3:- Data Formate of USMLE dataset.

1. **Prompt Engineering:** Which is different from actual fine-tuning. To get started, we don't need any technical knowledge or data. We can connect data through retrieval (RAG).
2. **Vector Databases:** We can use vectors for more storage for prompt engineering.
3. **Finetuning t :** Which include Instruction Tuning, Alignment Tuning, and Efficient Tuning. This teaches the model to behave more like a chatbot and creates a better user interface for model interaction.
4. **Finetune with RLHF:** We discuss it in further session in depth.
5. **Fine-tune with LOMO:** (LOw-Memory Optimization)

Let's dive into finetuning LLM techniques in more depth.

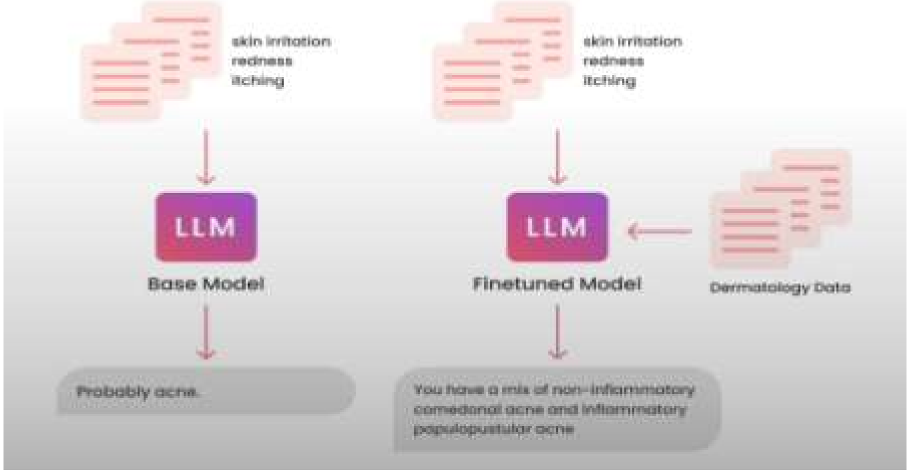


Figure 3.4:- Output difference between Base model and Finetuned model.

Parameter-Efficient Finetuningt (PEFT)

Traditional finetuning of pre-trained LLMs on downstream tasks yields significant performance gains. However, full finetuning becomes impractical due to model size and resource requirements [26]. Parameter-efficient finetuning (PEFT) methods address these challenges by finetuning only a small subset of model parameters. PEFT mitigates issues like catastrophic forgetting and improves performance in low-data and out-of-domain scenarios. PEFT methods are applicable across modalities and promote portability by generating smaller checkpoints. Various PEFT techniques include LoRA, Prefix Tuning, Prompt Tuning, and PTuning, with more to come. PEFT enables comparable performance to full finetuning with fewer trainable parameters. We can see different types of PEFT libraries in figure 3.5

QLora: Efficient Finetuning of Quantized LLMs

An effective finetuning method that maintains full 16-bit finetuning work speed while using adequate memory to fine-tune a 65B parameter model on a single 48GB GPU. Gradients are backpropagated into Low-Rank Adapters (LoRA) using QLoRA via a frozen, 4-bit quantized pre-trained language model [30]. That is seen in figure 3.7.

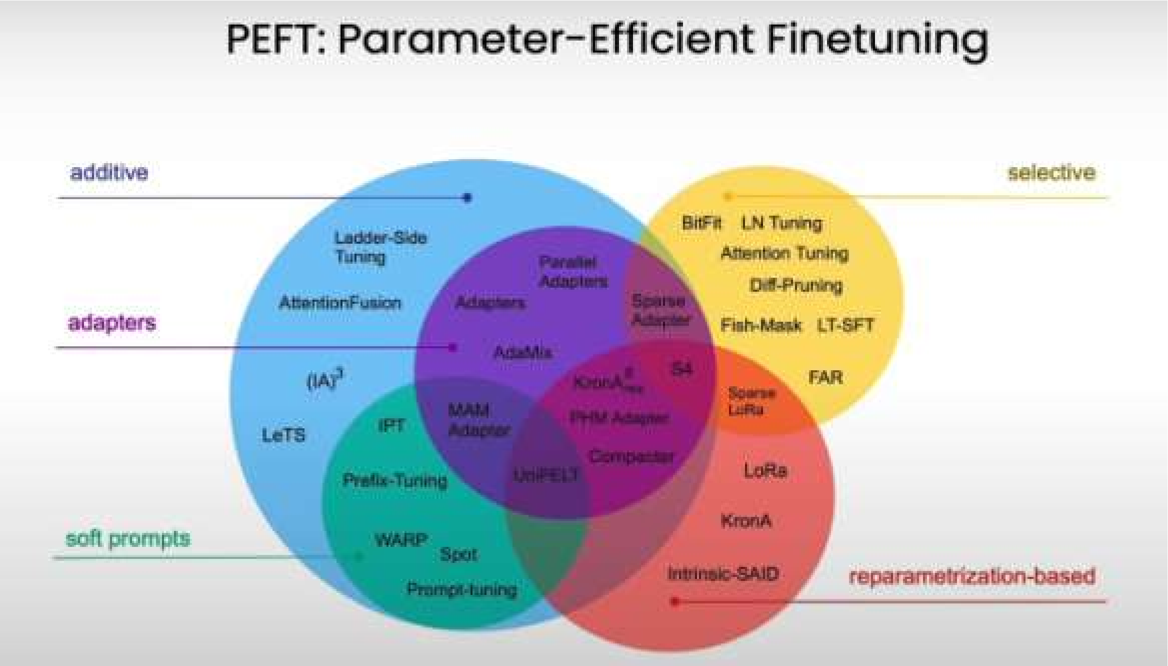
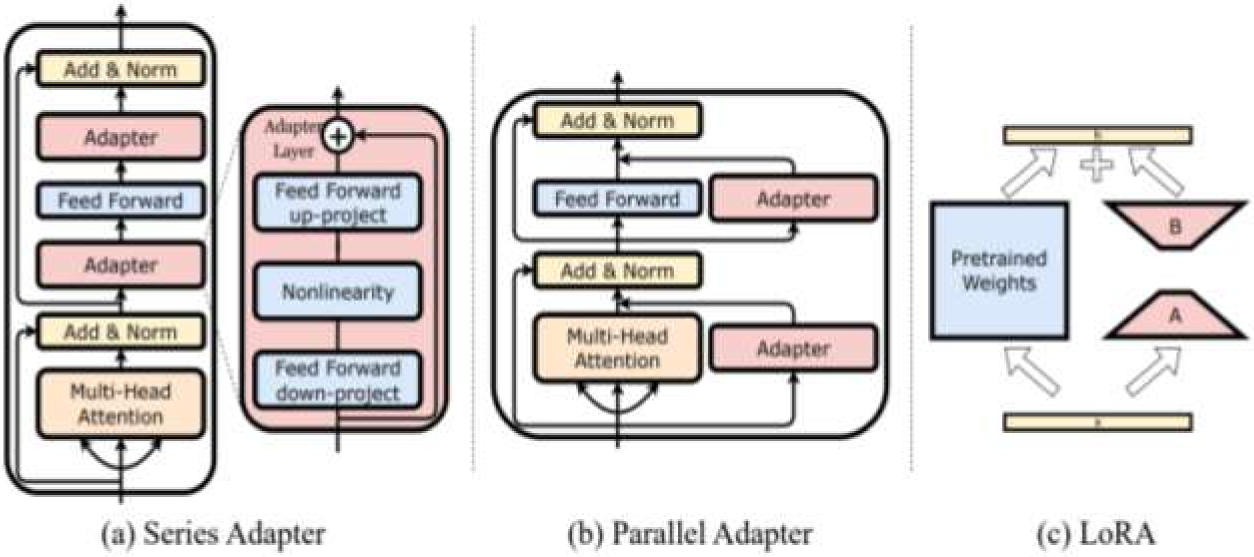


Figure 3.5:- PEFT: Parameter-Efficient fine tuning.



(a) Series Adapter (b) Parallel Adapter (c) LoRA

Figure 3.6:- LoRA: Low-Rank Adaptation of Large Language Models [26].

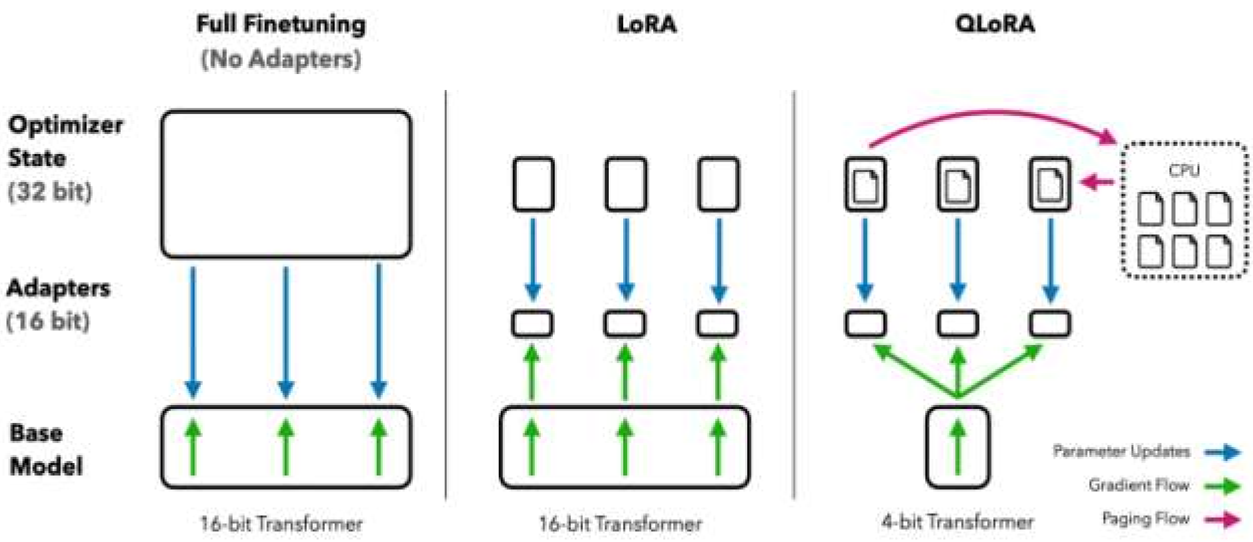


Figure 3.7:- Output difference between Base model and Finetuned model.

Several advancements are introduced by QLoRA to conserve memory without compromising performance: (a) A novel data type that is informationtheoretically ideal for normally distributed weights is 4-bit NormalFloat (NF4). (a) Using double quantization to lower the mean memory quant_config = BitsAndBytesConfig (load_in_4bit = **True** , bnb_4bit_use_double_quant = **True** , bnb_4bit_quant_type = " nf4 " , bnb_4bit_compute_dtype = torch . bfloat16)

Reinforcement Learning with Human Feedback (RLHF)

Strengthening Using human feedback data, Learning from Human Feedback (RLHF) refines large language models (LLMs) to produce models that are more in line with human preferences. RLHF guarantees that LLM resultsminimize any harm by staying away from offensive language and subjects, while maximizing utility and relevance to input requests. LLMs can be personalized by using RLHF, which allows models to continuously learn user preferences. Through actions in an environment and rewards or penalties based on the results, an agent learns to make decisions to accomplish a specified goal through reinforcement learning (RL), a type of machine learning.RLHF adapts RL concepts to the context of finetuning LLMs, where the LLM acts as the agent, the environment is the context window of the model, and the action generates text.

Rewards in RLHF are assigned based on how closely LLM completions align with human preferences, often evaluated against metrics such as toxicity. Obtaining human feedback for rewards can be time-consuming and expensive, so a reward model can be used as an alternative to evaluating LLM outputs against human preferences. The reward model is trained with human examples using supervised learning and then used to assess LLM outputs and assign reward values, which are used to update LLM weights iteratively. The reward model plays a central role in RLHF, encoding learned human preferences and guiding the model’s weight updates over iterations. We can see these processes in the figure 3.8

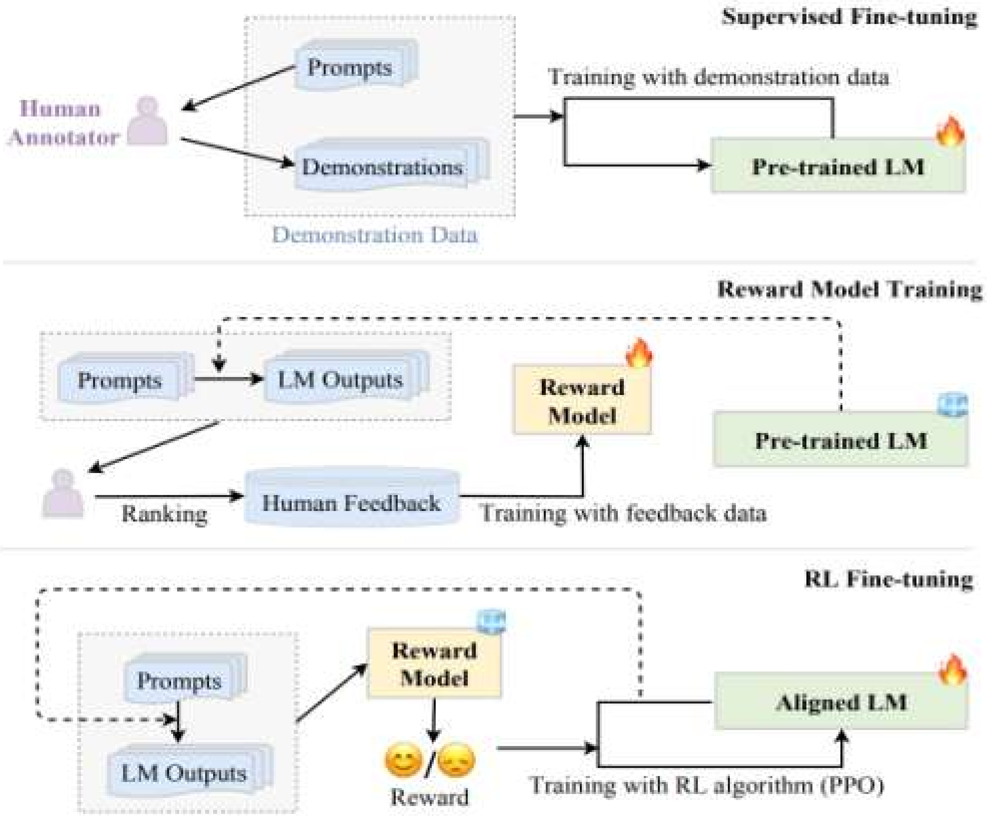


Figure 3.8:- Reinforcement Learning with Human Feedback (RLHF) cycle for Finetune LLMs.

Proximal policy optimization (PPO)
Proximal Policy Optimization (PPO) is a reinforcement learning algorithm that finetunes large language models (LLMs) towards human preferences. PPO updates the LLM policy through small, bounded changes over many iterations to ensure stability. PPO starts with an initial instruct LLM and goes through two phases: experimentation (Phase I) and policy update (Phase II), which is visible in figure 3.9

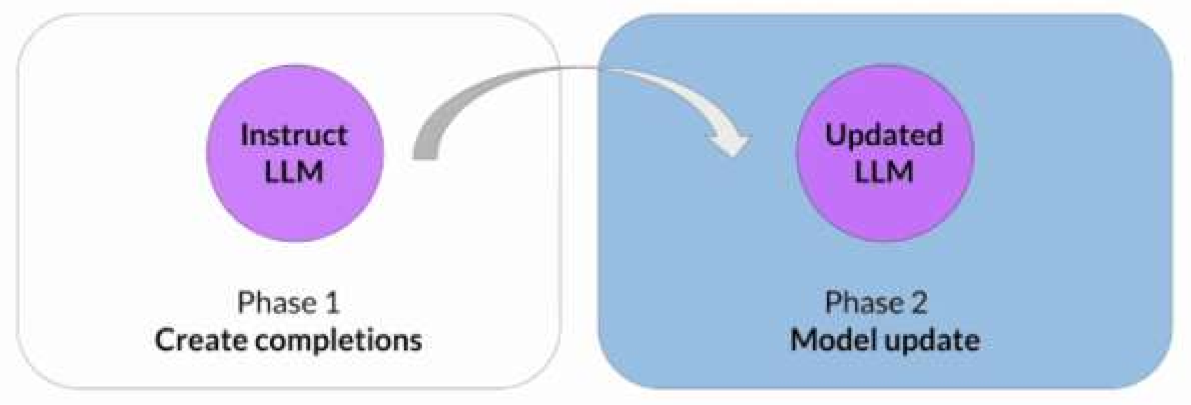


Figure 3.9:- PPO start with our initial instruct LLM and Generate RL-updated LLM.

In Phase I, the LLM completes prompts, and the reward model evaluates the completions based on human preferences. The value function estimates the expected total reward for a given state, helping evaluate completion quality against alignment criteria. Phase II involves updating model weights based on losses and rewards from Phase I while ensuring updates stay within a trust region [30].

The PPO policy objective aims to maximize the expected reward by updating LLM weights to produce more aligned completions. The policy loss, advantage estimation, and entropy loss are critical components of the PPO objective.

The PPO objective is a weighted sum of these components, stably guiding model updates towards human preferences. After several iterations, PPO results in a human-aligned LLM.

Other reinforcement learning techniques like Q-learning exist, but PPO is currently the most popular method due to its balance of complexity and performance. Research in finetuning LLMs through human or AI feedback is active, with new techniques like direct preference optimization (DPO) emerging.

Calculating Loss Finction

Calculating Value Loss:

Future reward predictions are more accurate as a result of the value loss. Phase 2 Advantage Estimation then makes use of the value function. This is comparable to when we begin writing a passage and already have a general notion of how it will turn out.In equation 3.1 L^{VF} is value loss.

L^{VF} [redacted] $V_\theta(s) -$ [redacted] (3.1)

Where,

S is a finite set of states, s_0 is an initial state, $\gamma \in (0, 1)$ is the discount factor, $r : S \rightarrow R$ is the reward function at given state,

$V_\theta(s)$ is Value function that estimates the future total reward.

- **Calculating Policy Loss:** This is where the proximal aspect of PPO comes into play, where the prompt completion, losses, and rewards guide model weights updates. PPO also ensures that the model updates within a small trust region. The PPO policy objective is the main ingredient of this method. Remember, the aim is to find a policy whose expected reward is high. In other words, we're trying to update the LLM weights that result in completions that align with human preferences and receive a higher reward.

LPOLICY [redacted] $\hat{A}t$, clip [redacted] ϵ, 1 [redacted]

Where, π_θ is model's probability distribution over tokens, a_t is the next token, s_t is the current state,

$\hat{A}_t$ is called the estimated advantage term of a given choice of action, epsilon is a hyperparameter.

- **Calculating Entropy Loss:** While the policy loss moves the model towards the alignment goal, entropy allows the model to maintain creativity. If we kept entropy low, we might always complete the prompt. Higher entropy guides the LLM towards more creativity.

L^{ENT} = entropy [redacted] (3.3)

Calculating Objective Finction

Our PPO target is the weighted total of all words, which steadily improves the model to reflect human preference. This is the PPO's overarching goal. The PPO goal uses backpropagation over a number of steps to update the model weights. PPO begins a new cycle after the model weights are modified. A new PPO cycle begins when the revised LLM is used in place of the old LLM for the subsequent iteration. You finally reach the human-aligned LLM after numerous iterations.

$$LPPO = LPOLICY + c_1LVF + c_2LENT \quad (3.4)$$

Where, c_1 and c_2 coefficients are hyperparameters.

Experimental Results:-

Accuracy

Accuracy gives us a straightforward understanding of how often the models generate the correct responses. It's a ratio of the accurate predictions to the total predictions made by the model. Here, accurate prediction means the correct option model will be chosen.

The silver standard will be shown in Table 4.1, which is 48%.

Table 4.1:- Evaluation of different LLMs on Zero-short Finetuning.

Model	Total Question	Correct Answer	Score(%)
Text_davici_003 Model	**100**	**48**	**48%**
Bloom_QLora_ft_MedMCQA_20K	100	28	28%
Bloom_QLora_ft_MedMCQA_20K_clean	100	38	38%
Mistral_7B_QLora_ft_MedMCQA_20K	100	45	45%
Bloom_QLora_ft_RLHF_MedMCQA	100	37	37%

None of the Above (NOTA) Test

In this test, the model has multiple-choice medical domain questions, and the correct answer is replaced by "None of the above." the model has to identify that option and justify its choice. The result of this experiment is shown with the Chain-of-Thought experiment setup.

prompt :

instruct : <instructions_to_llm > question : <medical_question > Options :
- 0: <option_0 >
- 1: <option_1 >
- 2: <option_2 >
- 3: <none_of_the_above > response :

cop : <correct_option > cop_index : <correct_index_of_correct_opt > why_correct :
<explanation_for_correct_answer > why_others_incorrect :
<explanation_for_incorrect_answers >

Chain-of-Thought prompting (CoT)

In CoT, the model is prompted to generate step-by-step solutions. CoT prompting led to substantial improvements in many reasoning-intensive tasks. It allows us to bridge the gap with human-level performances for most hard BIG-bench tasks [4]. As an alternative to writing reference step-by-step solutions, zero-shot CoT (Kojima et al., 2022) allows for generating CoTs using single and domain-agnostic cues: “Let’s think step by step”.

prompt for Zero−Short CoT: question : [Question]
Answer : Let ’ s think step by step <CoT>
Therefore , among the A through D, the answer is <answer>

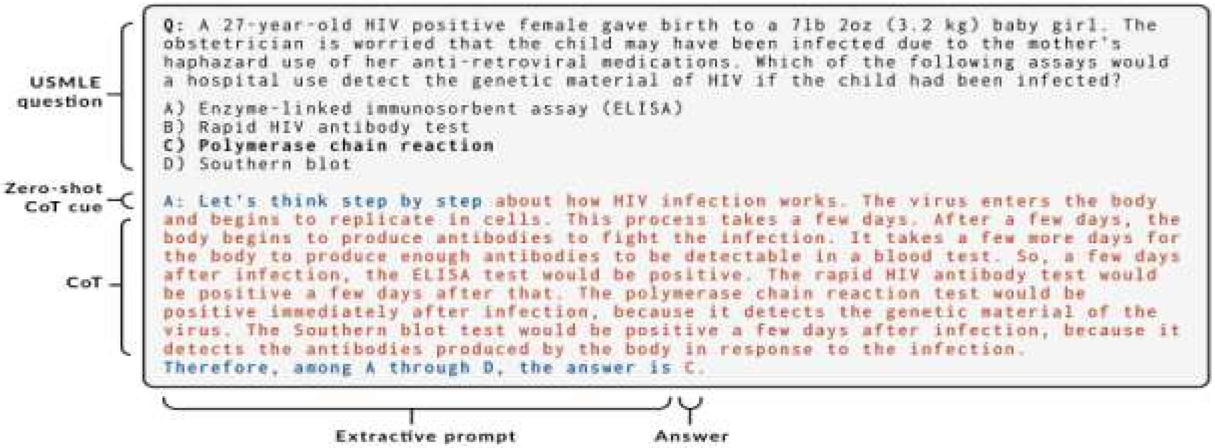
The following figure 4.1 shows the response of chain-of-thought prompting.

Table 4.2:- Evaluation of different LLMs on Zero-short CoT-Fine-Tuning.

Model	Total Question	Correct Answer	Score(%)	Increased (Points)
Text_davici_003 Model	100	53	53%	5
Bloom_QLora_ft_MedMCQA_20K	100	30	30%	2
Bloom_QLora_ft_MedMCQA_20K_clean	**100**	**48**	**48%**	**10**
Bloom_QLora_ft_RLHF_MedMCQA	100	43	43%	6

CoT prompting with Ensemble model

In this part of the experiment, we compare the completions z^1, . . . , z^k can be sampled from the generative LLMs. As the figure A.1 shows, we aggregate the completions and estimate the marginal answer likelihood.



Q: A 27-year-old HIV positive female gave birth to a 7lb 2oz (3.2 kg) baby girl. The obstetrician is worried that the child may have been infected due to the mother's haphazard use of her anti-retroviral medications. Which of the following assays would a hospital use detect the genetic material of HIV if the child had been infected?

A) Enzyme-linked immunosorbent assay (ELISA)
B) Rapid HIV antibody test
C) Polymerase chain reaction
D) Southern blot

A: Let's think step by step about how HIV infection works. The virus enters the body and begins to replicate in cells. This process takes a few days. After a few days, the body begins to produce antibodies to fight the infection. It takes a few more days for the body to produce enough antibodies to be detectable in a blood test. So, a few days after infection, the ELISA test would be positive. The rapid HIV antibody test would be positive a few days after that. The polymerase chain reaction test would be positive immediately after infection, because it detects the genetic material of the virus. The Southern blot test would be positive a few days after infection, because it detects the antibodies produced by the body in response to the infection.
Therefore, among A through D, the answer is C.

Figure 4.1:- Answering a USMLE question using zero-shot CoT prompting.

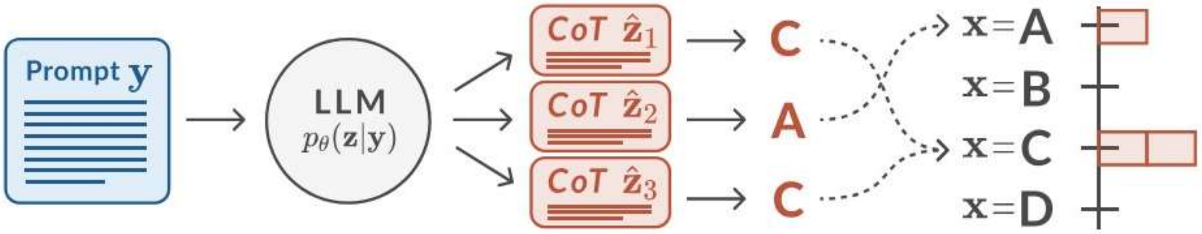


Figure 4.2:- Generative process and answer likelihood (ensemble model, i.e., selfconsistency).

In equation 4.1, x is the answer string, y is the prompt string, and z is a completion generated by LLM denoted by p_θ.

k

p[redacted], $z\hat{}_1$, .., $z\hat{}_k$ [redacted] (4.1)

i=1

Table 4.3:- Evaluation of different LLMs on Zero-shot CoT-Fine-Tuning with Ensemble Model.

Model	Total Question	Correct Answer	Score(%)	Increased (Points)
Text_davici_003 Model	100	53	53%	0
Bloom_QLora_ft_MedMCQA_20K	100	30	30%	0
Bloom_QLora_ft_MedMCQA_20K_clean	**100**	**52**	**52%**	**4**
Bloom_QLora_ft_RLHF_MedMCQA	100	46	46%	3

Conclusion:-

This study has shown how to modify large language models (LLMs) to create a medical domain-specific question-answering system. The suggested method makes use of open-source LLMs in tandem withusing fine-tuning methods like QLoRA and Parameter-Efficient Fine-Tuning (PEFT), which allow high-performing models to be deployed on common hardware with little computational expense. Additionally, by bringing model outputs into line with human expectations, Reinforcement Learning with Human Feedback (RLHF) produces responses that are more dependable and appropriate for the given environment. The results show that Chain-of-Thought (CoT) prompting and ensemble techniques, in conjunction with smaller, domain-adapted LLMs, can greatly improve performance on medical text-based tasks.. Future work may focus on advancing fine-tuning methodologies and expanding system capabilities to address more complex and nuanced medical queries. This progress will not only improve human-AI interaction but also enable more trustworthy decision-support systems. Additionally, by incorporating Retrieval-Augmented Generation (RAG) techniques into prompt engineering, it is possible to further elevate reasoning accuracy, ultimately aiming to approach or match the performance of state-of-the-art models such as OpenAI's GPT-3 Davinci.

References:-

1. Author, F.: Article title. Journal 2(5), 99–110 (2016). Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.
2. Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... Natarajan, V. (2022). Large language models encode clinical knowledge. arXiv preprint arXiv:2212.13138.
3. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... Wang, H. (2023). Retrievalaugmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
4. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... Zhou, D. (2022). Self-consistency improves the chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.
5. J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 2023.
6. Zhang, J. Sun, Y. Du, and H. Zhang, "FinGPT: Financial Large Language Models," arXiv preprint arXiv:2210.12345, 2022.
7. T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, 2020.
8. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal-BERT: The muppets straight out of law school," arXiv preprint arXiv:2010.02559, 2021.
9. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.
10. T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLaMA-INT8: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2212.09820, 2022.
11. Vaswani, A., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 5998–6008.
12. Bommasani, R., et al. (2021). On the Opportunities and Risks of Foundation Models. arXiv:2108.07258.
13. Brown, T., et al. (2020). Language Models are Few-Shot Learners. In NeurIPS, 33, 1877–1901.
14. Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
15. Yang, Z., et al. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS.
16. Raffel, C., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR, 21(140), 1–67.
17. Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.
18. BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model.
19. Technology Innovation Institute. Falcon LLM: Open-source LLMs for production environments.
20. Meta AI. LLaMA 2: Open foundation and fine-tuned chat models.
21. Dettmers, T., et al. (2022). "LoRA: Low-Rank Adaptation of Large Language Models", UW NLP Group.
22. EleutherAI. GPT-NeoX-20B: An open-source 20B parameter autoregressive language model.
23. Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilic´, S., ... Bari, M. S. (2022). Bloom: A 176b-parameter open-access multilingual language model.arXiv preprint arXiv:2211.05100.
24. Xiao, A., Zheng, W., Keren, G., Le, D., Zhang, F., Fuegen, C., ... Mohamed, A. (2021). Scaling ASR improves zero and few-shot learning. arXiv preprint arXiv:2111.05948.
25. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I. (2023, July). Robust speech recognition via large-scale weak supervision. In International conference on machine learning (pp. 28492-28518). PMLR.
26. Xu, L., Xie, H., Qin, S. Z. J., Tao, X., Wang, F. L. (2023). Parameter-efficient finetuning methods for pre-trained language models: A critical review and assessment. arXiv preprint arXiv:2312.12148.
27. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
28. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.
29. Dataset-USMLE from MedQA by Jin, Di et al. "What disease does this patient have? a large-scale open domain question answering dataset from medical exams." https://github.com/jind11/MedQA
30. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., ... Christiano, P. F. (2020). Learning to summarize with human feedback. Advances in Neural Information Processing Systems, 33, 3008-3021.