الجمهورية الشعبية الديمقراطية الجزائرية
People's Democratic Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry of Higher Education and Scientific Research
جامعة باتنة 2 مصطفى بن بولعيد
University of Batna2
Mostefa Ben Boulaïd

Master's Thesis

To obtain the diploma of **Master's Degree**

Field of Study: **Computer Science**

Specialization: **Artificial Intelligence & Multimedia**

Theme

Building a Trustworthy Medical Assistant Using Fine-Tuned LLMs and Retrieval-Augmented Generation

---

Presented by
Abderrahmen Melkemi
Abdelmalek Ounis

Defended on: **Mai,2025**
In front of the jury composed of

Dr. Abderrezak BENYAHIA	President of the Jury
Dr. Badreddine Benreguia	Master's Thesis Supervisor
Prof. Ouahab KADRI	Examiner

Academic Year: 2024/2025

Acknowledgement

Life was always a long road, one that we can never walk alone, while achievements are those high mountains those that we could never reach or climb without help. That being said, we would like to express endless gratitude and thanks to:

First and foremost, God, the Almighty, for the countless blessings those which we know about and those we don't.

We would also like to express our sincere gratitude to our supervisor, Dr. Badreddine Benrguia, Head of the Department of Computer Science at the University of Batna2, for providing us with a conducive academic environment and excellent facilities throughout our Master's program in Multimedia and AI. Additionally, we are immensely thankful to Dr. Benrguia for granting us the opportunity to work on this captivating Medical Chatbot project, which has significantly enriched our knowledge in this field.

Finally, we would like to acknowledge our families and friends for their unwavering motivation and encouragement during the course of our project work. Their support has been instrumental in our success.

Abstract

The integration of Large Language Models (LLMs) into healthcare presents significant opportunities for improving clinical decision-making, patient communication, and medical information retrieval. However, the direct deployment of state-of-the-art models such as GPT-4, Claude, or Gemini in real-world clinical environments faces critical limitations. These include concerns over data privacy due to reliance on third-party APIs, high computational requirements that restrict accessibility in resource-limited settings, and the risk of generating inaccurate or unverifiable medical content commonly referred to as hallucinations. This thesis proposes the design and development of a secure, efficient, and locally deployable medical assistant system based on open-source LLMs. The approach centers on fine-tuning DeepSeek-R1-Distill-Qwen-7B using QLoRA (Quantized Low-Rank Adaptation), a memory-efficient method that enables training on limited hardware through 4-bit quantization and adapter modules. To enhance the factual accuracy of generated responses, the system integrates a Retrieval-Augmented Generation (RAG) pipeline, which grounds model outputs in verified medical documents retrieved via vector-based semantic search. Domain-specific fine-tuning with clinical datasets and instruction-driven data improves language specialization and reduces hallucinations.

The resulting system demonstrates that competitive performance in medical NLP tasks can be achieved without relying on proprietary infrastructure or excessive computational resources. The project contributes a scalable, auditable, and ethically aligned proof of concept that advances the democratization of medical AI, making it accessible for deployment in hospitals, research centers, and educational institutions.

Keywords— Deep Learning, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Chatbot, Hallucination , Fine-Tuning, Intelligent Medical Assistant.

LIST OF ACRONYMS

AI : Artificial Intelligence
LLM : Large Language Model
NLP : Natural Language Processing
RAG : Retrieval-Augmented Generation
CNN : Convolutional Neural Network
RNN : Recurrent Neural Network
LSTM : Long Short-Term Memory
GAN : Generative Adversarial Network
BERT : Bidirectional Encoder Representations from Transformers
BPE : Byte Pair Encoding
API : Application Programming Interface
QLoRA: Quantized Low Rank Adaptation
PEFT : Parameter-Efficient Fine-Tuning

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

Nowadays, the rise of artificial intelligence (AI), particularly through advancements in deep learning, has profoundly transformed the field of natural language processing (NLP) [21]. At the forefront of this transformation are large language models (LLMs), which represent a significant leap in AI capabilities [5]. Built upon complex deep learning architectures such as the Transformer [41], these models have demonstrated exceptional performance across a wide range of NLP tasks, including text classification, summarization, machine translation, conversational agents, and complex question answering. Their ability to understand, generate, and interact in human language has unlocked new possibilities for deployment in high-stakes domains, most notably in medicine [38].

The healthcare sector increasingly looks for the incorporation of AI-powered systems to complement clinical decision-making, assist medical professionals, and facilitate access to information for patients as well as practitioners. Of these systems, LLMs stand out for their capacity to provide rapid, contextually relevant answers to medical inquiries, compile vast amounts of scientific literature, and offer user-friendliness. The deployment of such systems in clinical settings remains minimal due to several main challenges.

All the most sophisticated LLMs such as GPT-4, Claude, and Gemini are proprietary and based on cloud-based infrastructure. This gives rise to serious data protection and security concerns, especially in medical environments where patient data confidentiality is regulated very strictly. Offloading sensitive medical data to third-party APIs is not feasible for hospitals and clinics owing to ethical, legal, and technical constraints. Besides, engaging with third-party vendors puts institutions under commercial licensing, internet connectivity, and fluctuating cost models.

The deployment of LLMs requires gigantic computational power. Training and even inference with large models often require the application of powerful GPU clusters and massive memory, something which is typically out of reach in the majority of hospitals or schools, particularly in low income developing nations. Such hardware requirements pose a gargantuan barrier to entry.

LLMs have also been reported to produce hallucinations well formed in terms of syntax but wrong in facts [16]. In the medical world, they have dangerous implications in the form of false diagnoses, therapeutic suggestions, or misinterpretation of symptoms. This limits the dependability of LLMs and poses safety problems that have to be solved before they can be applied to real world healthcare.

To address these pressing issues, this master's project proposes the creation and deployment of a secure, locally deployable medical assistant system powered by an open source LLM. The goal is to deliver sophisticated language models into healthcare organizations without exposing patient data, requiring enormous budgets, or relying on third party APIs.

The core of such a system is a finely tuned language model that has been fine tuned using a memory efficient approach named QLoRA (Quantized Low Rank Adaptation) [8]. QLoRA applies 4-bit quantization and LoRA adapters to enable fine-tuning of large models in small hardware setups, for instance, in a single or dual GPU setup. This significantly lowers the compute costs without any trade-off in the model's performance. Along with this, we also suggest a new fine-tuning method

involving domain-instruction tuning, edited samples of real medical conversations, and customized loss functions that mitigate hallucinations and factual inaccuracy.

We also introduced a Retrieval Augmented Generation (RAG) mechanism that enhances response confidence by grounding model responses in an external knowledge base of verified medical content [23]. When a user enters a query, the RAG system retrieves highly relevant documents using vector-based semantic search (e.g., FAISS), and the documents are used as contextual input for the model. This method ensures the generated response consists of up-to-date and medically verified information, rendering the process unbiased and transparent.

The result is an affordable, interpretable, and hybrid model within the operational constraints of health facilities. It shows that aggregated and fine-tuned small and medium sized open source LLMs, when supported by document retrieval, can achieve competitive performance in domain related areas like medicine without the need for resource intensive or proprietary models.

Ultimately, this work contributes to the democratization of medical AI. By challenging the limitations of current LLMs through new technical solutions and with an emphasis on ethical, secure deployment, we offer a deployable solution that can be scaled, audited, and trusted by doctors.

This thesis is structured as follows:

- **General Introduction** presents the motivation behind the project and the challenges of deploying LLMs in healthcare.
- **Chapter 1** presents the technical background, discussing foundations of large language models, Transformer architecture, fine-tuning approaches including QLoRA, and principles of Retrieval-Augmented Generation (RAG).
- **Chapter 2** introduces the research problem and outlines the scientific and technical objectives, including limitations of LLMs in healthcare.
- **Chapter 3** details the implementation of the proposed medical assistant, covering environment, datasets, fine-tuning pipeline, integration of QLoRA and DeepSpeed, and RAG design.
- **Chapter 4** reports the experimental evaluation with performance metrics, hallucination detection, system behavior, and comparison with and without RAG.
- The **Conclusion** summarizes the contributions, implications for deployment, and future directions.

CHAPTER 1

BACKGROUND

1.1 Introduction

Artificial Intelligence (AI) is evolving rapidly, profoundly transforming several sectors, including healthcare. Among the most promising technologies, Large Language Models (LLMs) stand out for their ability to understand and generate human language in a fluent and contextual manner. Their application in the medical field is generating significant interest due to their potential to automate documentation, assist in diagnosis, and facilitate access to clinical knowledge.

LLMs thus deserve particular attention, as they combine advances in Deep Learning and Generative AI to offer powerful tools that require a thorough understanding of their functioning, advantages, and limitations.

1.2 Deep Learning and the Evolution of AI

1.2.1 Deep Neural Networks

Deep Learning is based on deep neural networks capable of modeling complex relationships within data. These architectures are inspired by the human brain and are composed of interconnected layers that enable hierarchical abstraction of data [12].

Convolutional Neural Networks (CNNs) have led to major advances in computer vision, while Recurrent Neural Networks (RNNs) have been applied to sequence processing, particularly in language tasks.

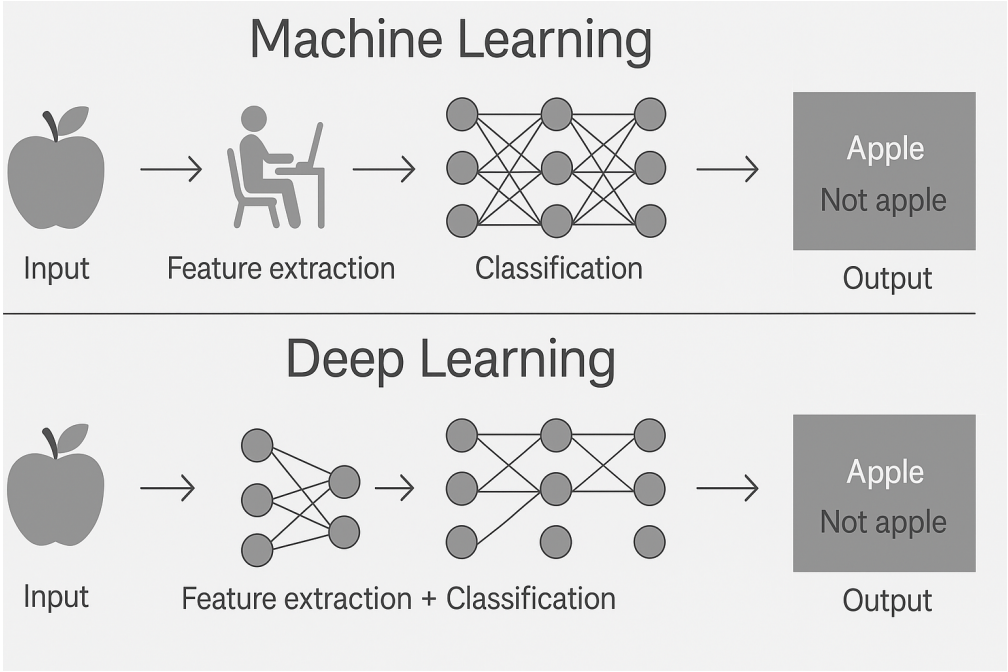

Figure 1.1: The difference between machine learning and deep learning.

1.2.2 Major Advances in Deep Learning

The introduction of the Transformer architecture by Vaswani et al. in 2017 marked a major breakthrough [41]. It enables parallel processing of sequences while effectively capturing long-range dependencies through the attention mechanism.

Models such as BERT [9], GPT [31], T5 [32], and LLaMA [40] are based on this architecture and now dominate natural language processing (NLP).

Table 1.1: Examples of LLMs and their key characteristics.

Model	Year	Size
BERT	2018	340M
GPT-2	2019	1.5B
GPT-3	2020	175B
LLaMA 2	2023	7B to 65B

1.2.3 Limitations of Classical Models

Before the Transformers, RNNs and LSTMs were commonly used in NLP, but these architectures had several limitations:

- Inability to efficiently handle long-term dependencies.
- Sequential training that is slow and difficult to parallelize.
- Gradient-related problems (exploding or vanishing gradients).

Traditional models also required extensive feature engineering and lacked generalization capabilities. Transformers, combined with large datasets and powerful computational resources, have enabled the training of generative models capable of solving diverse tasks with few or no examples.

1.3 The Emergence of Generative AI

1.3.1 Definition of Generative AI

Generative Artificial Intelligence (Generative AI) refers to a branch of AI whose main objective is to create new content from existing data. Unlike discriminative models, which learn to classify or predict, generative models learn the underlying distribution of data and can produce new, similar instances [12].

This content can include text (as with LLMs), images, audio, videos, or even source code. This capability makes generative AI especially powerful in creative, medical, educational, and industrial domains.

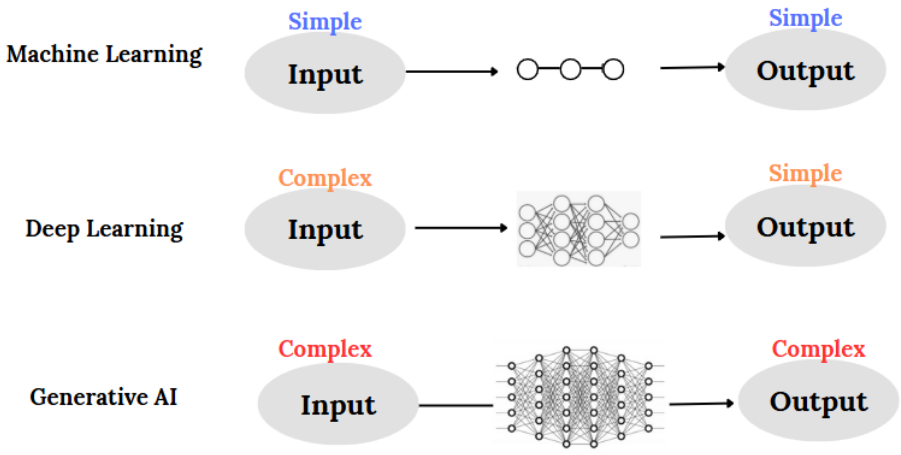

Figure 1.2: The difference between machine learning, deep learning and Generative AI.

1.3.2 Generative Adversarial Networks, Diffusion Models and LLms

Several approaches have emerged for content generation:

- **Generative Adversarial Networks (GANs)**: Introduced by Goodfellow et al. in 2014, GANs consist of two neural networks (a generator and a discriminator) competing in a zero-sum game [11]. They are particularly effective at generating realistic images.
- **Diffusion Models**: More recent, these models generate images by learning to progressively denoise random data. They are the foundation of tools like DALL · E 2 and Stable Diffusion [13].
- **LLMs (Large Language Models)**: Based on the Transformer architecture, LLMs such as GPT-3 or LLaMA generate natural language text. They are trained on very large text corpora and use techniques such as next-token prediction to generate coherent content [5].

1.3.3 General Applications of Generative AI

Generative AI is now deployed in many domains:

- **Healthcare**: Generation of medical reports, assistance in prescription writing, simulation of clinical cases.
- **Education**: Automatic creation of lessons, exercises, or interactive content.
- **Creative industries**: Generation of images, music, videos, and even narrative scripts.
- **Software development**: Tools like GitHub Copilot offer AI-assisted code completion.

1.3.4 Risks and Challenges of Generative AI

Despite its progress, generative AI presents several challenges:

- **Truthfulness and hallucinations**: Models like GPT can generate inaccurate or fabricated information.
- **Ethics and bias**: AI can reproduce biases present in the training data [3].
- **Malicious uses**: Creation of deepfakes, spam, or automated misinformation.
- **Intellectual property**: Generating content based on existing data raises legal questions.

It is therefore essential to regulate the development of these technologies to ensure they remain ethical, safe, and beneficial.

1.4 Large Language Models

1.4.1 Overview

LLM (Large Language Model) refers to a specific kind of neural network that is specifically designed for natural language processing tasks such as language understanding and language generation. "Large" in the case of LLM refers to the humongous number of parameters (typically in the billions) and astronomical datasets upon which the model has been trained to identify with so that it can pick up nuanced patterns in human language.

Google researchers introduced an NLP revolution in 2017 with their paper "Attention Is All You Need," where they presented the Transformer architecture. The new architecture attempted to solve some of the issues of previous Seq2Seq (Sequence-to-Sequence) models, which were generally based on RNNs or LSTMs. These previous models had difficulty handling long input sequences, especially because information from previous points in the sequence would tend to get lost during backpropagation, through issues such as the vanishing gradient problem [42].

Transformers addressed this by employing a mechanism known as self-attention, which allows the model to focus on different parts of the input sequence at the same time, regardless of their position. This idea was originally inspired by the attention mechanism introduced by Bahdanau et al. in 2014, but the Transformer took it much further by removing recurrence altogether and relying solely on attention.

Later, in October 2018, Google published BERT (Bidirectional Encoder Representations from Transformers), a major breakthrough in language understanding. BERT employed only the encoder part of the Transformer architecture (as opposed to the original Transformer, which had a decoder as well). The large version of BERT, having 350 million parameters, was the largest Transformer-based model at that time. It achieved state-of-the-art performance in most NLP tasks and laid the groundwork for most LLMs that have been developed since.

1.4.2 Main Components of an LLM

1.4.2.1 Tokenization

Tokenization is the initial process carried out before sending data to a large language model (LLM). In this, text is divided into very small pieces called tokens and mapped to numeric values like token IDs. This is because LLMs do not process raw text; they process numeric representations only.

There are different methods of tokenization, and any specific LLM may use a particular one of its preference depending on the vocabulary size of the data for which it is trained and the model architecture. By coincidence, there are largely three types of tokenization which are:

- **Word-based Tokenization:** This is the simplest and most general method. It splits text based on word spaces. For example: "The doctor works" → ["The", "doctor", "works"]

  But this method has one major drawback: it leads to a huge vocabulary due to different forms of the same word. For instance, the word "work" can be "works", "worked", "working", etc., and each of them can be treated as a different token.

  As a result, the vocabulary size can be 500,000+ tokens, and the model becomes harder to train. And if we try to reduce the vocabulary size, we will have lots of <UNK>tokens (unknown words), i.e., the model does not know lots of important words — and that is a critical problem for LLMs since it affects accuracy and understanding.

- **Character-based Tokenization:** In this approach, each character is treated as an individual token and numbered. For example: "cat" → ["c", "a", "t"]

  This does make the vocabulary very small (just letters, punctuation, etc., say 100 tokens altogether), but it makes the input sequence very large, especially for longer bits of text, which means that the model must consider an astronomical number of tokens, making it more computationally expensive and slower.

  So while it avoids the <UNK>problem, it's not suitable for most purposes, especially in large models.

- **Byte Pair Encoding (BPE):** This is a very popular subword tokenization technique, and it's used in most modern LLMs like GPT-4, LLaMA, DeepSeek-R1, and so on. It combines the strengths of the previous techniques. Here's how it works:

  - It starts with a first vocabulary of single characters.
  - Next, it joins the most frequent pairs of tokens or characters to form subwords.
  - Over time, common patterns like "ing", "tion", or even whole words are acquired as new tokens.

  This results in:

  - Lower vocabulary than word-based methods.
  - Fewer <UNK>tokens, because low-frequency words can still be broken up into familiar subwords.
  - Tokens with a balanced length — not too long like character-based, and not too massive like word-based.

The Table 1.2 below represent an example of (BPE) on words "word", "world", "east", "waste". where the symbole $< /w >$ represnt the end of word:

Pair	Frequency
w o	3
o r	2
r d	1
d </w>	2
o l	1
l d	1
e a	1
a s	2
s t	2
t </w>	1
w a	1
t e	1
e </w>	1

Table 1.2: Byte Pair Encoding Pair Frequencies example

1.4.2.2 Embedding Layer

In this step, token IDs are passed to your LLM (Large Language Model). When the tokens are first passed into the model, they undergo a process called embedding. This embeds the tokens by a neural network to convert each token into a vector embedding, which captures token features and represents it as a vector. Each vector dimension has a set of objects that have certain features. This way, the embedding gives semantic meaning to each word, encoding its context-specific meaning and mapping it to a multi-dimensional space. The resulting vector representation holds the word's meaning, syntactic, and semantic subtleties that are crucial in understanding language.The Figure 1.3 reprenset the result of embbeding process.

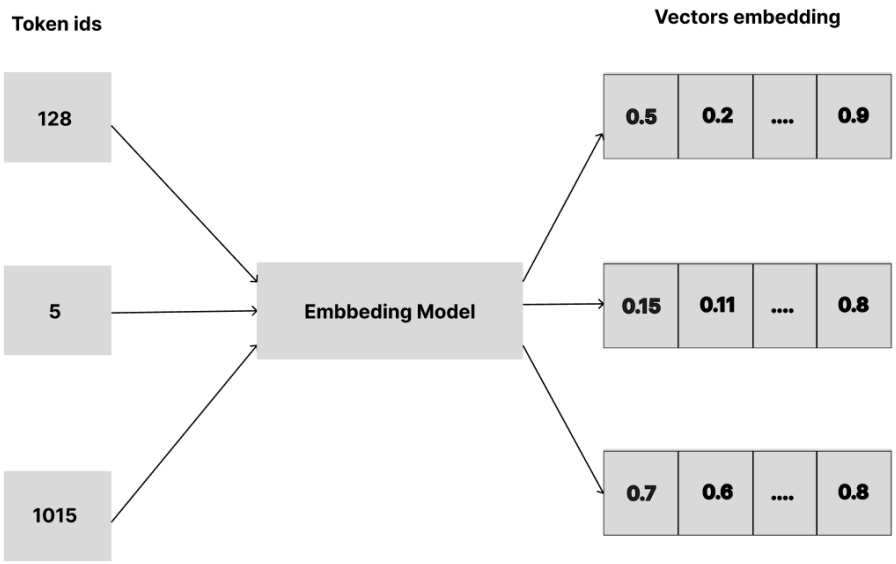

Figure 1.3: illustration of process of embbeding layer

1.4.2.3 Transformer Architecture

Transformer is the core engine and fundamental part of large language models (LLMs). After the paper "Attention Is All You Need," Figure 1.4, it comprises two broad categories: an encoder, which processes and understands the input through multiple encoding layers that operate on the embedded vectors, and a decoder, which generates the output based on the encoder's output through decoding layers to fine-tune and develop the final output.

The encoding layer consists of two sublayers: multi-head self-attention and a feed-forward network. While the decoding consists of masked multi-head attention, multi-head attention, and a feed-forward network[42].

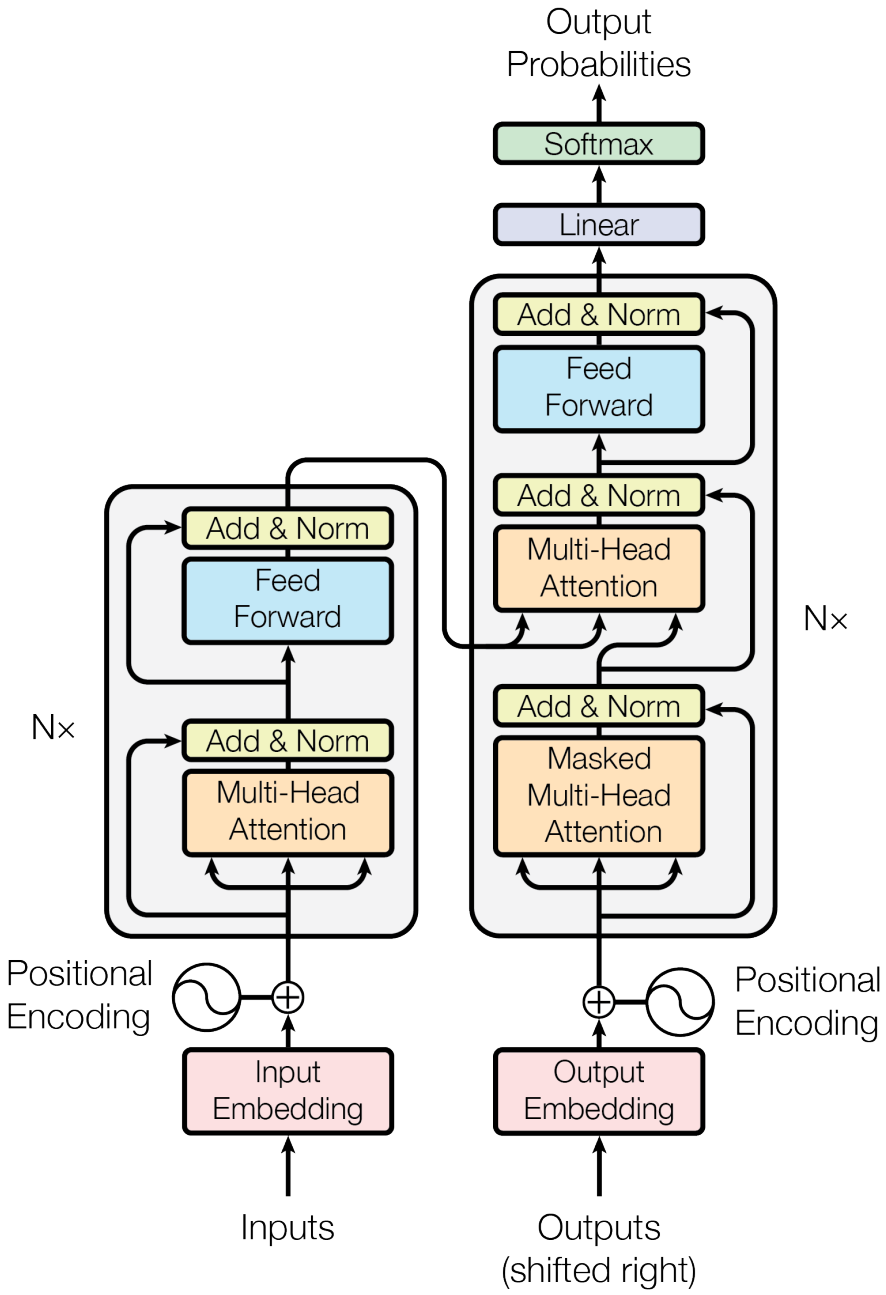

Figure 1.4: The Transformer - model architecture (source:"Attention Is All You Need" paper)

Multi-head self-attention: Its primary function is to capture and model the dependencies between tokens, regardless of their distance in the input sequence. The mechanism utilizes three matrices: Wq (query), Wk (key), and Wv (value). A set of these matrices together forms an attention head. The Figure 1.5 represent the attention dot prodect.

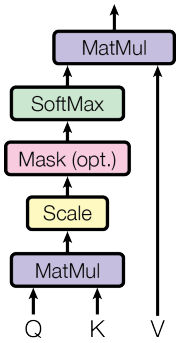

Figure 1.5: The Transformer - model architecture (source:"Attention Is All You Need" paper)

While the Figure 1.6 represent Multi-Head Attention.

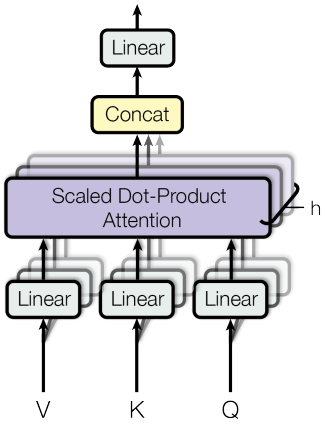

Figure 1.6: Multi-head self-attention (source:"Attention Is All You Need" paper)

The attention mechanism is characterized by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where:

- Q = Query matrix
- K = Key matrix
- V = Value matrix
- d_k = Dimension of the keys

Feed-forward network: It is a two-layered multi-perceptron network where:

$$FFN(x) = max(0, xW_1 + b1)W_2 + b2$$

Where:

- W_1 and W_2 are weight matrices
- b_1 and b_2 are bias vectors.

Masked attention: It restricts the model to look only at previous and current tokens when computing attention.

1.4.2.4 Positional Encoding

This phase provides tokens with positional information. Since LLMs, unlike RNNs, read tokens in sequence in parallel and hence don't inherently understand the token order, the issue arises. Positional encoding is introduced to tackle this problem by providing information about each token's position within the sequence. This allows the model to take into account the word order and relative positions, which is crucial in learning the sentence structure and meaning.

Sinusoidal Functions: This method uses pre-established sine and cosine functions, which are added to embedding vectors to preserve positional information. The idea is to encode positions based on mathematical patterns which the model can be trained to read. The sinusoidal positional encoding is represented by:

$$PE_{(pos,\,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

$$PE_{(pos,\,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

Where:

- **pos** is the position of the token in the sequence,
- **i** is the dimension index,
- $d_{\textbf{model}}$ is the model's hidden size (i.e., the size of the embedding vector).

These functions ensure that each position has a unique representation and that similar positions have similar encodings, which helps the model learn about word order and token relationships.

1.4.2.5 Output Layer

The output of the transformer is a sequence of token logits numerical scores corresponding to the probability of each token in the tokenizer's vocabulary. In the output layer, the model predicts the next token by applying the softmax activation function to these logits, converting them into a probability distribution.

Since LLMs predict text one token at a time, the model takes the last predicted token as input for the next step. This continues until a stopping criterion is reached (e.g., a maximum number of tokens or an end-of-sequence token).

In addition, the output behavior can be guided by sampling strategies that influence the generation of the final text:

Temperature Scaling:

- This technique alters the "confidence" of the predictions made by the model.
- It does this by dividing the logits by a temperature T > 0 before softmax.
  - A temperature that is lower (<1) will result in the model being more confident and deterministic.

– A temperature that is higher (>1) will result in more randomness and creativity.

- Formula:

$$softmax(logits/T)$$

Top-k Sampling

- This method is an enhancement of temperature sampling by only looking at the top k highest logits (most likely tokens).
- All other token probabilities are masked (set to zero) to discourage the model from picking low-probability (typically incorrect) tokens.
- After masking, optional temperature scaling is done, and softmax is performed to sample from the top k.
- This balances diversity with coherence in generated text.

1.4.3 Training Process of Large Language Models

There are two main steps that Large Language Models (LLMs) are based on:

1.4.3.1 Pretraining

The goal is to train the model to learn to build meaningful and coherent text. The model is trained to forecast the following token in a sequence using an enormous amount of unlabeled text, such as books, articles, and web pages. This phase is highly resource- and time-intensive, as it involves learning the overall form and patterns of language.

1.4.3.2 Fine-tuning

Fine-tuning is concerned with adapting the pretrained model to carry out a particular task, like question answering or summarization. This process employs a smaller, task-specific dataset and demands much fewer resources than pretraining, which makes it more efficient and cost-effective.

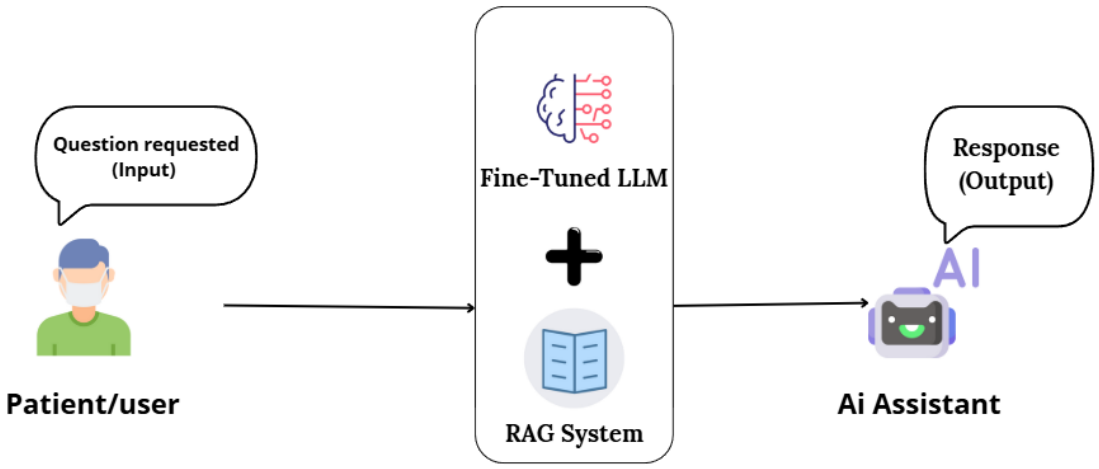

Figure 1.7: Overview of the proposed medical assistant: a fine-tuned open-source LLM augmented with a medical knowledge retrieval system.

1.4.4 Retrieval-Augmented Generation for LLMs

1.4.4.1 Overview of Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by dynamically integrating external knowledge during generation. Unlike standard LLMs that rely solely on parametric memory (Section 1.4.2), RAG combines:

- **Parametric knowledge**: Learned during training (Section 1.4.3)
- **Non-parametric knowledge**: Retrieved from external sources

This architecture addresses key LLM limitations [23]:

- Static knowledge cutoff dates
- Factual hallucinations
- Lack of source attribution

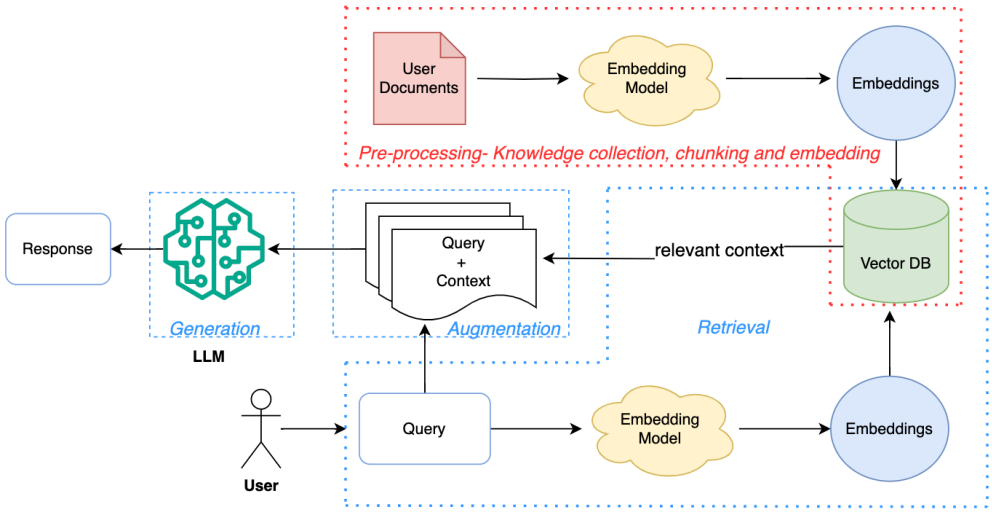

Figure 1.8: RAG architecture showing the retriever-generator interaction. Adapted from [23].

1.4.4.2 Architecture and Components

RAG's two core components integrate with standard LLM architecture (Section 1.4.2):

1. **Retriever**:
    - Encodes queries using the same embedding layer (Section 1.4.2.2)
    - Uses dense retrieval (DPR [20]) or sparse retrieval (BM25)
    - Outputs top-k relevant documents
2. **Generator**:
    - Augments transformer attention (Section 1.4.2.3) with retrieved context

Component	LLM Integration Point
Retriever	Embedding layer (Section 1.4.2.2)
Generator	Cross-attention layers (Section 1.4.2.3)

Table 1.3: How RAG components connect to standard LLM architecture

- Implements either:
  - RAG-Sequence: Single document for full generation
  - RAG-Token: Dynamic document switching

1.4.4.3 Applications of Retrieval-Augmented Generation Systems

Notable implementations include:

- Open-domain QA systems [**?**].
- Enterprise knowledge management.
- Real-time information systems.

1.5 The Goal of LLMs

The goal of LLMs is to create models that can handle a diverse set of NLP tasks with minimal task-specific adjustments, providing an all-encompassing solution to language understanding and generation.

1.5.1 Iconic Models

Several LLMs have emerged over the years, each contributing uniquely to the field of NLP. Below are some of the most notable models:

- **GPT (Generative Pre-trained Transformer)**: Developed by OpenAI, the GPT models are pre-trained on large text corpora and then fine-tuned for specific tasks. GPT-3, for example, has 175 billion parameters and has demonstrated remarkable capabilities in text generation, code writing, and more [31].
- **BERT (Bidirectional Encoder Representations from Transformers)**: Created by Google, BERT focuses on bidirectional context, allowing the model to understand a word's meaning based on its surrounding context. BERT has been highly effective in tasks like question answering, sentence classification, and named entity recognition [9].
- **LLaMA (Large Language Model Meta AI)**: Developed by Meta, LLaMA aims to deliver high performance while maintaining efficiency in training and inference. It is designed to compete with models like GPT and has achieved competitive results in NLP tasks [40].
- **DeepSeek**: A specialized LLM that excels in domain-specific tasks, such as medical applications or other niche knowledge areas, making it suitable for industry-specific use cases.
- **Claude**: Named after Claude Shannon, this model is a competitor to GPT-3, focusing on generating coherent and contextually appropriate text with fewer parameters.

1.5.2 Size, Training, Datasets, and Costs

The performance of LLMs is largely determined by their size (i.e., the number of parameters), the quality and diversity of the datasets used for training, and the computational resources available. Larger models tend to outperform smaller ones across a wide range of tasks, but they also require exponentially more data and computational power to train.

For example, GPT-3, which has 175 billion parameters, was trained on a vast array of text data from books, articles, and websites [5]. Training such large models is costly, with estimates placing the total cost of training GPT-3 at several million dollars. Additionally, the environmental impact of training large models is a growing concern due to the high energy consumption associated with the necessary computational resources.

1.5.3 Hallucinations and Limitations of LLMs

Despite their impressive capabilities, LLMs have notable limitations, with one of the most significant issues being the phenomenon of "hallucinations." Hallucinations occur when an LLM generates text that is factually incorrect or entirely fabricated but appears plausible to the reader. This is particularly problematic in high-stakes domains like healthcare, law, or finance, where incorrect outputs can have serious consequences.

Other limitations of LLMs include:

- **Bias**: LLMs can inherit and even amplify biases present in the training data, leading to potentially harmful or unfair outputs. Addressing bias in LLMs is an ongoing research challenge [3].
- **Overfitting**: While large models excel at general tasks, they may struggle with specialized domains without further fine-tuning. Overfitting can occur when the model's performance on training data does not generalize well to real-world applications.
- **Computational costs**: Training and deploying LLMs require substantial computational resources, which can be a barrier for smaller research labs, companies, and organizations.

1.6 LLMs in the Medical Field

1.6.1 Use Cases: Medical Assistants, Documentation, Triage

Large Language Models (LLMs) are increasingly being used in the medical field to enhance various aspects of healthcare. A common use case is the deployment of medical assistants powered by LLMs, which can interact with doctors and patients by providing relevant answers to medical questions. These assistants can also help with medical documentation, such as automatically generating summaries of consultations, prescriptions, or medical reports.

Additionally, LLMs can be used for patient triage, assisting in prioritizing the most urgent cases based on the described symptoms and available medical information, thus reducing the workload of healthcare professionals.

1.6.2 Advantages: Accessibility, Reducing Workload, Clinical Support

The use of LLMs in the medical field offers several advantages. Firstly, it improves accessibility to healthcare, particularly in rural areas or contexts where qualified healthcare professionals are scarce. By providing real-time support, LLMs assist doctors, aid decision-making, and provide relevant and up-to-date information.

Moreover, LLMs reduce the cognitive and administrative burden on doctors by automating repetitive tasks such as data entry or document creation. This enables healthcare professionals to focus more on the clinical and human aspects of their work while benefiting from technical assistance for more mechanical tasks.

Finally, LLMs provide clinical support by offering information based on the latest medical research and helping manage complex diseases, especially in fields with a high volume of data.

1.6.3 Risks: Errors, Hallucinations, Ethics

Despite their advantages, LLMs present significant risks in the medical field. One major concern is the risk of errors generated by the model, especially when it produces incorrect or inappropriate responses. Due to their tendency to generate "hallucinations" — plausible-sounding but factually inaccurate information — these models can mislead users, particularly in critical contexts.

Another concern is ethical in nature. The use of LLMs to make medical decisions raises questions about responsibility in case of errors, the protection of patient privacy, and the impact of automation on the doctor-patient relationship. It is crucial to ensure that LLMs adhere to ethical standards and regulations regarding the privacy and security of medical data.

1.6.4 Importance of Explainability and Verifiability

One of the major challenges of using LLMs in the medical field is the explainability of their decisions. Large language models are often seen as "black boxes," making it difficult to understand the reasoning behind a decision or why certain information was generated. This lack of transparency is especially problematic in a sensitive field like medicine, where decisions must be justified and verifiable.

Thus, it is essential to develop methods to make LLMs more explainable so that doctors can understand and validate the results provided by these models. Moreover, mechanisms for verification must be established to ensure that the information generated by LLMs is accurate, relevant, and in line with medical best practices.

1.7 Conclusion

In this chapter, we explored the use of LLMs in the medical field, highlighting their use cases, advantages, and associated risks. We also discussed the importance of explainability and verifiability in the medical context, emphasizing the challenges these models face when deployed in sensitive environments.

In conclusion, while LLMs hold significant potential to improve healthcare, it is essential to account for the risks associated with their adoption and implement regulatory and oversight mechanisms to ensure their responsible and ethical use. The following chapter will focus on related work in this field and the challenges posed by the integration of these models into healthcare systems.

CHAPTER 2

PROBLEM STATEMENT AND RESEARCH OBJECTIVES

2.1 Introduction

The last few years have seen unprecedented growth in the artificial intelligence domain, spearheaded notably by the advent of Large Language Models (LLMs) [41]. ChatGPT-4 (OpenAI), Claude (Anthropic), and DeepSeek (DeepSeek AI Lab) are a few of the models that have shattered previous machine learning limitations in natural language processing and generation. Their ability for fluent, contextual dialogue has enabled innovative applications ranging from customer service automation to specialized legal research support.

Of the potential areas of application, healthcare is both the most critical and most promising. Consider scenarios where a physician consults an AI model in real time for diagnostic recommendations or a nurse utilizes automated patient note generation to free up time. These scenarios are quickly becoming realities. LLMs have the potential to transform healthcare by facilitating real-time clinical decision-making, democratizing medical knowledge, and enhancing patient communication.

Yet this promise is accompanied by tremendous challenges. The same models capable of generating coherent medical guidance are also able to generate incorrect or misleading information. The opacity of these models raises concerns regarding trust and transparency. Ethical concerns emerge as these technologies start to impact decisions under which human lives are at stake. Furthermore, the absence of alignment with established clinical guidelines hinders their implementation in routine medical care.

In a profession where accuracy is everything and errors can have life-altering consequences, even minor errors can have extreme and long-lasting impacts [4]. This thesis addresses these critical challenges, aiming to advance the trustworthiness and safety of LLMs for medical use.

2.2 Defining the Research Problem

Large Language Models (LLMs) have exhibited impressive natural language processing abilities with the potential for revolutionary applications in medicine. Nevertheless, several key challenges restrict their deployment in real clinical environments. These challenges constitute the central research problems that this thesis seeks to address:

- **Factual Hallucinations:** LLMs can generate factually incorrect yet coherent medical information, posing serious risks to diagnosis, treatment, and patient safety. Such hallucinations undermine trust and reliability [16]. While some approaches attempt to reduce hallucinations, they generally cannot guarantee medically accurate outputs in critical scenarios.

- **Lack of Access to Verifiable and Updated Sources:** Most LLMs are trained on static datasets and lack real-time access to verified and updated medical knowledge. Retrieval-Augmented Generation (RAG) methods show promise to address this limitation, but securely integrating live medical databases remains challenging [23].
- **Opacity and Insufficient Transparency:** The internal decision-making processes of LLMs are largely opaque, impeding clinician trust. Although explainability methods exist, they often fall short in high-stakes healthcare contexts [36].
- **Ethical, Legal, and Privacy Issues:** Patient data require stringent protection under regulations such as HIPAA and GDPR. Cloud-based deployments raise privacy concerns, and federated learning has been explored but faces scalability challenges [37].
- **High Computational and Resource Requirements:** Large LLMs demand substantial computational resources, limiting accessibility in healthcare. Parameter-efficient fine-tuning methods like LoRA and QLoRA show promise but are not yet widely adopted in medicine [8].

These challenges emphasize the urgent need for methods that improve the accuracy, safety, interpretability, and accessibility of LLMs in healthcare. The following chapters will explore potential solutions and methodologies designed to tackle these core issues.

2.3 Challenges in Applying LLMs in Healthcare

2.3.1 Factual Hallucinations and Clinical Risks

Large Language Models (LLMs) have long been known to produce*hallucinations*-output that appears confident and self-assured despite being factually incorrect. Such hallucinations in the clinical context can have severe consequences. For instance, ChatGPT-4 has been found to deliver misleading drug interaction data which, if used, could lead to serious side effects [4].. Similarly, Claude, another widely used LLM, has been found to be unreliable when summarizing clinical studies, occasionally attributing therapeutic effects to medications without adequate supporting evidence [5].

Such errors are especially dangerous in advisory or diagnostic applications, where users may place high confidence in the model's responses. One incorrect recommendation could delay treatment, prescribe inappropriate or harmful therapy, or even encourage dangerous self-medication practices [38, 22]. In the high-stakes setting of clinical environments, minimizing factual hallucinations is critical to ensuring patient safety and maintaining clinical integrity.

Figure 2.1 demonstrates a common instance of LLM hallucination in the medical domain. While the recommendations sound rational and are presented in slick medical terminology, they are factually inaccurate and not grounded in clinical practice.

This vagueness becomes critically problematic in recommendation or diagnostic applications, where end-users cannot be guaranteed by the model output. One bad recommendation could lead to delaying treatment, prescribing an unsafe or ineffective medication, or encouraging risky self-treatment behaviors [38, 22]. Because of the high-stakes environments of clinical settings, minimizing fact-based hallucinations is necessary to ensure patient safety as well as the credibility of the clinic.

Figure 2.1: Example of hallucinations in a medical context related to Type 1 diabetes. LLMs may generate incorrect medical recommendations that seem plausible but are factually inaccurate.

Real-World Example

In my own testing,We posed a question to ChatGPT-4 in medicine, i.e., "What does a purple ESR result indicate?" The model responded as follows:

> Purple ESR result most likely refers to the color of the blood tube and not the result itself. Let me spell it out for you.
>
> What is 'Purple ESR'? The purple (or lavender) top tube in lab tests contains EDTA, the anticoagulant for the majority of blood tests–to wit:
>
> - Complete Blood Count (CBC)
> - Erythrocyte Sedimentation Rate (in some procedures)
>
> Despite this, in most cases, ESR is performed using a sodium-containing black-top tube with citrate in most nations. Some laboratories utilize the EDTA (purple) tubes according to chemical modulation.
>
> This option might be feasible because it exactly specifies the color of the tube for the ESR testing.

However, the question was about exactly what a "purple ESR result" would be, and ChatGPT-4 incorrectly used the color of the collection tube, not the test result.

The answer should have been:

> Erythrocyte Sedimentation Rate (ESR) is a laboratory test that determines the rate at which red blood cells descend in a tube over a specified period of time. The amount is quantified in millimeters per hour (mm/hr), not in colors. Misperception usually results from the test tube color used for specimen collection, but it does not affect the reported results. This shows how apparently reasonable but wrong data can be an indicator.

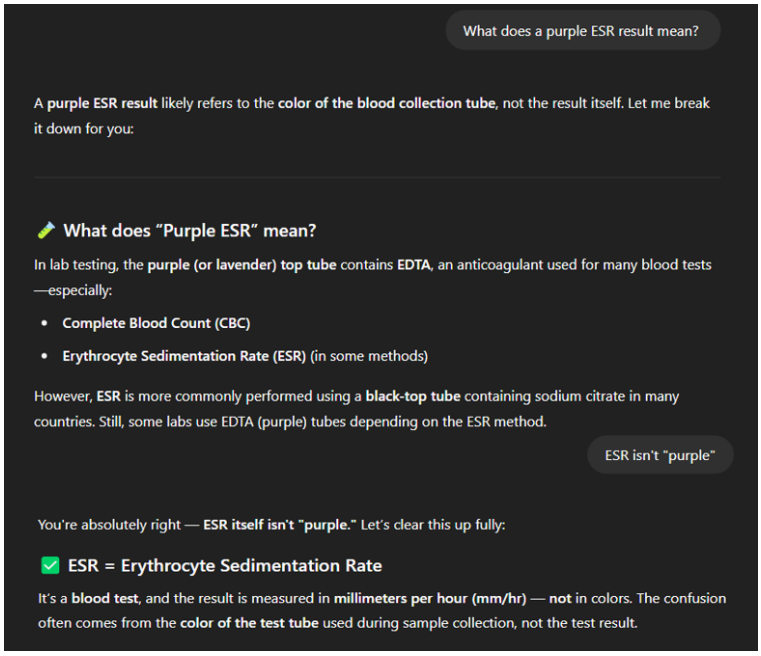

Figure 2.2: Real-world example of hallucinations in a medical context. ChatGPT-4 misinterpreted the question and provided an incorrect response regarding the meaning of a "purple ESR result."

We can't afford to accept such results at face value without verification. (See Figure 2.2)

The increased stakes of the clinical situation remind how we need to address risk in order to improve patient safety. Accessing LLM can support better medical practice, but in order to enhance trustworthiness of LLM-based support, few steps need to be taken, including *real-time verification*, *cross-referencing* with trusted medical databases, and continuously updating and training data.

Statistics on Hallucinations in LLMs:

Table 2.1 provides some statistics of hallucinations within LLMs, in particular with ChatGPT-4, Claude, and DeepSeek. These statistics shed light on the frequency and nature of hallucinations in real-world practice, with a clear indication to use LLMs cautiously in high-risk areas, or to mediate in some ways the use of LLMs in clinical care.

The statistics clearly show that hallucination rates with different models will differ as well as with different domains. Statistics for unconsidered risks urges action to implement better verification and safety measures to mitigate risks associated with use of LLMs in applied high-stakes work like health care and beyond.

Model	Hallucination Rate	Domain	Common Issue
ChatGPT-4	15-30%	Healthcare	Drug interactions, treatment recommendations
Claude	10-25%	Healthcare	Clinical research summaries, therapeutic effects
DeepSeek	5-10%	Healthcare	Diagnostic support, medication dosages

Table 2.1: Statistics on hallucination rates in different LLMs. The values represent reported rates of hallucinations in healthcare-related queries.

2.3.2 Lack of Verifiable Sources

Modern-day large language models (LLMs) do not normally quote sources or provide citable trails. For a person working clinically, there is a need for evidence-based information and peer-reviewed literature, an essential limitation. In a clinical sense, there is utility to LLM-generated responses

but the clinician cannot know or track where the information is coming from or weigh the quality of the guidance provided [6]. The lack of origin tracking reduces the utility of these tools in healthcare and raises questions about safe onboarding and fit in clinical practice.

The inability to trace the origin of medical assertion creates a divide in clinical practice. For example, if the LLM proposes an off-label drug usage pattern without quoting any central studies or guidelines, the advice might seem reasonable yet is not traceable in a clinical way. In that case, the clinician cannot validate the output and could be directed to pursue inappropriate or dangerous strategies especially in high-risk domains like the emergency department, oncology, or intensive care.

Verifiability is the foundation of evidence-based medicine (EBM) because it is based on transparent, peer-reviewed, and reproducible evidence. LLMs cannot be used in this way and therefore violate EBM's premises – and patients are in jeopardy – most importantly – a real risk of eroding clinician's trust – alongside surmountable obstacles to their use across large swathes of applying health care in workflows.

We have only just begun to engage with citation systems for LLMs well being LLM Chatbots such as ChatGPT-4 are mainstream as this article now goes live, the issues of attribution of the sources are as probative with real-world application. This is also a very daunting consideration within the context of a clinical setting where any account should be corroborated with the benefit of a witness of honest standing. The study led by [26] already observed that, as best practice, if using AI-based medical information to do so in conjunction with verifiable medical references. Healthcare can also utilize the same process and the authors of this study recommend regular visits and source-tracing validation for credibility and social acceptance across clinical AI systems. Here they define a system that can reduce hallucinations by almost 37

Model	**Hallucination Rate**	**Accuracy**
ChatGPT-4	High	Moderate
Claude	Moderate	High
DeepSeek	Low	High

Table 2.2: Comparison of hallucination rates, source citation practices, and model accuracy in popular LLMs. The absence of verifiable sources remains a major barrier to the clinical integration of these tools.

2.3.3 Outdated Knowledge and Static Training Data

One of the major shortcomings of existing Large Language Models (LLMs), including ChatGPT-4, is that they are based on static training data which is limited to a knowledge cutoff date—usually one to two year prior to deployment. This temporal distribution leaves LLMs vulnerable in rapidly changing fields such as health care where an updated knowledge base is critical. Current LLMs cannot access or reason with new clinical guidelines, recent approvals for medications or therapies, or the cutting-edge evidence that is being revealed through ongoing clinical trials [31].

Even progressive LLMs such as DeepSeek and Claude who are trained using multilingual and domain-specific habits cannot overcome this limitation unless appropriately fine-tuned or augmented with retrieval-augmented systems. This concern is even more pronounced in rapidly evolving medical fields such as oncology, infectious diseases, and pharmacology, where treatment protocols can shift significantly and a matter of days or weeks can mean the difference between carrying forward a relevant new guideline and being stuck with outdated or potentially harmful recommendations.

The static nature of LLM knowledge degrades the confidence and trust of health care professionals in their outputs; for example, we know practitioners expect assurances that model recommendations stay up to date with existing standards of care.

2.3.4 Overconfidence and Misinformation

One of the primary issues with LLMs in healthcare is that they appear to provide answers with high levels of confidence even when the response is speculative, incomplete, and inaccurate. In a clinical context, this overconfidence can be dangerous because effective care and patient safety depend on accurate, evidence-based recommendations. Existing models like Claude provided highly confident information even when presenting speculations within their field of expertise. Claude has stated many rare diseases and experimental therapies with seemingly little if any, evidence of solid clinical studies or peer-reviewed research [24]. This could mislead both healthcare professionals and patients especially when the consumer of the content does not have the ability to critically evaluate the content. In the context of clinical practice this type of behavior can lead to unfortunate consequences. Healthcare professionals may sometimes trust a model's output and proceed with treatment plans based on inaccurate or unverifiable information. In other instances, a patient may self-diagnose or self-treat with information that they simply read - with reasonable confidence that the information is valid - potentially harming themselves or others. Such risks become exponentially severe in high stakes scenarios; for example, the management of a rare disease, plans for a complex surgery, or prescribing medicines characterized as high risk medications. Additionally, given that LLMs use a conversational style, it is likely researchers, healthcare professionals and patients will not receive the cautious skepticism that could help moderate overconfidence. Therefore, addressing overconfidence is a key part in safely incorporating LLMs into healthcare environment without compromising patient care [27].

2.3.5 Ethical and Legal Implications

When considering the ethical and legal dilemmas when using Large Language Models (LLMs) for clinical decision making, accountability is perhaps the most pressing challenge: when an AI assistant makes a recommendation that leads to harm, who is responsible? There is an evolving legal landscape when it comes to AI and healthcare with no definitive legal expectations on liability. When adverse outcomes are triggered by an AI system recommendation, identifying who is responsible (the developers and the healthcare provider or the user) is a challenging task [45].

In addition to concerns regarding accountability, the use of LLMs raises numerous concerns regarding patient privacy and informed consent. LLMs may predispose users to divulge protected medical data when they write their prompts, but unless the LLM has strong privacy protections,that possible loss of privacy could be a violation of patient privacy laws such as the General Data Protection Regulation (GDPR) in Europe or the Health Insurance Portability and Accountability Act (HIPAA) in the United States. Both laws intend to safeguard patient confidentiality and whether medical information via LLM is retained and handled with caution and security. With no data protection protocol, LLMs may unintentionally profane patients' medical privacy, creating risk of violating the law and reducing public trust too.

The opacity in LLMs output generation complicates informed consent as well. Both patients and health professionals using these models may not fully understand the implications and risks of employing AI to make clinical decisions, or even the data that trained the models. This could result in a patient or healthcare provider issuing consent as a result of medical advice based on either faulty or biased information, with potentially dire consequences. To address these ethical and legal issues, it is essential to have strong regulatory frameworks,transparency in AI decision-making processes, and clear accountability structures. Data safeguarding measures, transparency measures, and establishing accountability will be paramount for the safe incorporation of LLMs in healthcare settings.

Challenge	Ethical Concern	Legal Implications
Accountability	Who is responsible for harm caused by AI decisions?	Undefined liability in current legal frameworks
Privacy	Sensitive data sharing and lack of safeguards	Violation of GDPR, HIPAA, and other privacy laws
Transparency	Lack of understanding of AI's decision-making	Informed consent challenges and legal complications

Table 2.3: Ethical and legal challenges associated with using LLMs in healthcare. These issues highlight the need for clear guidelines, legal accountability, and robust privacy protections.

2.3.6 Bias and Inequity in Medical Recommendations

Large Language Models (LLMs) are trained on extensive datasets that often reflect historical and societal biases. In the medical domain these biases can have significant consequences. When models generate recommendations that vary based on patient race,gender or socioeconomic status despite identical clinical profiles—this introduces inequity in care and can exacerbate existing disparities in healthcare systems [10].

For example, biased chatbots outputs may suggest different treatment paths or risk factors simply based on demographic cues. This behavior undermines principles of fairness and personalized medicine and poses ethical concerns, especially if such tools are integrated into clinical workflows.

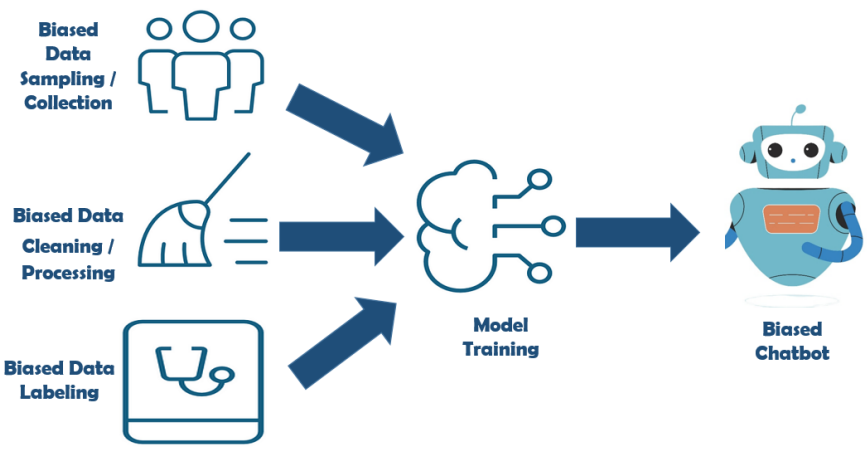

Figure 2.3: Illustration of the creation process of a biased chatbot. Biases in data sampling, cleaning, and labeling feed into the training process, resulting in outputs that can perpetuate unfair or discriminatory medical recommendations.

Inconsistencies like this goes beyond just being technical errors; there are legal and ethical implications. Discriminatory outputs from AI in healthcare can result in unequal treatment, erosion of patient trust, and breaches of anti-discrimination laws. If LLMs are not diligently audited and mitigated, they also run the risk of actively worsening systemic bias as opposed to improving health equity in healthcare.

As a result, bias in LLMs is a significant issue and must be dealt with before these systems can be safely implement in a high-stakes environment like healthcare.

2.4 Research Objectives

With the above-mentioned major challenges of applying LLMs to healthcare, this thesis also aims to provide solutions making the models safer, more reliable, and more domain-specific. The key

research goals are:

- **Reduce factual hallucinations using Retrieval-Augmented Generation (RAG):** Construct and implement a RAG pipeline that combines pre-trained LLMs with external, fact-checked medical knowledge bases such that outputs are informed by accurate, up-to-date information, taking the challenge of hallucinations and misinformation head-on [23].
- **Increase transparency and trust through explainability and source traceability:** Establish means of providing clinicians with intelligible outputs supported by traceable citations from the sources of the retrieved medical content, responding to the opaqueness and lack of transparency of LLM decisions [34].
- **Decrease clinical risks of hallucinations:** Systematically compare hallucination rates in clinical LLMs across benchmark sets and utilize fine-tuning and prompt engineering techniques to reduce clinically dangerous mistakes, thus enhancing patient safety [16].
- **Specialize LLMs for medical use via efficient fine-tuning:** Leverage parameter-efficient fine-tuning methods such as QLoRA for fine-tuning general-purpose LLMs to the medical task while evading the generalization limitations and boosting performance on clinical specialized tasks [14].

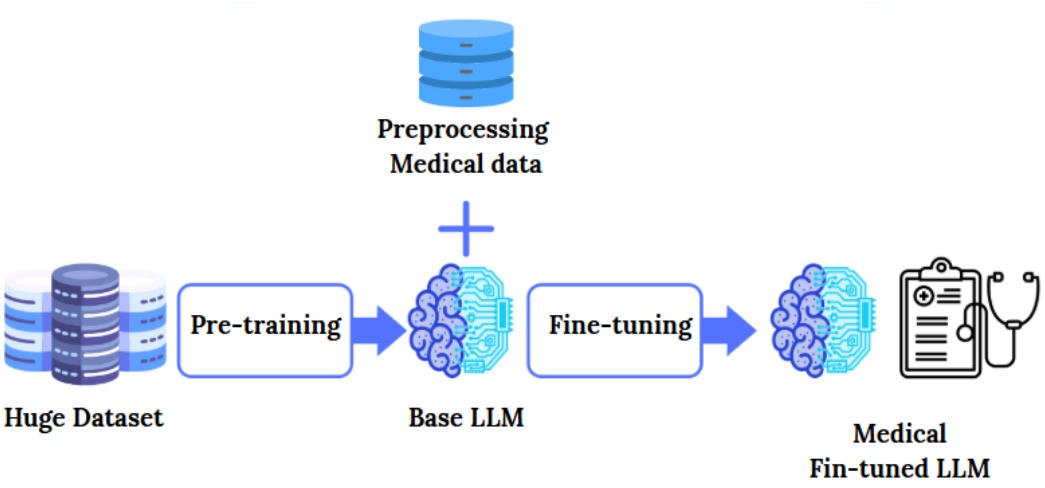

Figure 2.4: Fine-tuning large language model

- **Streamline computational requirements to increase accessibility:** Investigate quantization and light fine-tuning techniques for reducing hardware requirements to enable deployment of LLMs in typical health care settings where processing capacity is limited [8].

2.5 Scope and Limitations

This research focuses on the application of Large Language Models (LLMs) to clinical decision support systems using text-based data and tasks. The scope is limited to medical question answering and retrieval tasks on structured and unstructured textual data. It does not extend to other modalities such as medical imaging, audio, or real-time clinical deployment.

The work is restricted to open-source models and publicly available datasets due to licensing constraints and accessibility limitations. As a result, experiments are conducted in controlled offline

settings, with evaluation criteria centered on factual accuracy, relevance, and explainability of the generated responses, rather than clinical effectiveness in real-world scenarios.

The following factors have influenced the scope and scale of this study:

- **Hardware Constraints:** Limited access to high-performance computational resources restricted the scale of model training, inference, and experimentation.
- **Data Limitations:** High-quality, domain-specific medical datasets are rarely available as open-source. This limited the ability to fine-tune or evaluate models on clinically rich and representative data.
- **Time Constraints:** Given the complexity of designing, implementing, and evaluating LLM-based systems, time was a major constraint on iterative development, optimization, and extensive experimentation.

Despite these limitations, this thesis proposes practical approaches to improve the accuracy, safety, and domain alignment of LLMs in healthcare through Retrieval-Augmented Generation and efficient fine-tuning techniques.

2.6 Conclusion

This chapter has clarified and discussed the core problems that arise when employing Large Language Models (LLMs) in healthcare, from factual hallucinations, source unverifiability, stale knowledge, ethical concerns, and high resource consumption of these systems. This chapter has also clarified the research objectives that seek to address these issues, as well as the scope and limitations that demarcate the boundaries of this work.

By defining the problem of research and its proposed solutions, this chapter establishes the tone for the rest of the thesis. The following chapter will provide the theoretical foundation by discussing the history of deep learning and the development of generative AI technologies that support contemporary LLMs.

CHAPTER 3

METHODOLOGY AND SYSTEM DESIGN

3.1 System Design

To address the critical challenges of accuracy, security, and accessibility in medical AI, this project proposes the design and deployment of a secure, locally operable medical assistant system based on an open-source large language model (LLM). The objective is to integrate advanced language models into healthcare environments without exposing sensitive patient data, relying on third-party APIs, or requiring prohibitively expensive infrastructure.

At the heart of the system is a fine-tuned LLM adapted with QLoRA (Quantized Low-Rank Adaptation), a memory-efficient fine-tuning technique that applies 4-bit quantization and LoRA adapters. This enables powerful model customization on modest hardware, such as a single or dual-GPU setup, significantly reducing computational costs while maintaining high performance. Fine-tuning is further enhanced through domain-specific instruction tuning, curated examples from real medical dialogues, and specialized loss functions to reduce hallucinations and increase factual reliability.

To ensure responses are grounded in trustworthy knowledge, the system incorporates a Retrieval-Augmented Generation (RAG) architecture. This mechanism retrieves relevant documents from a curated medical knowledge base using FAISS-based semantic search and feeds them as context to the LLM during response generation. This approach strengthens the factual accuracy of outputs and improves transparency by anchoring answers in verified medical literature.

The resulting system is a cost-effective, explainable, and hybrid AI solution designed to function within the operational limits of healthcare institutions. It demonstrates that well-fine-tuned, open-source LLMs—when paired with robust retrieval systems—can deliver competitive performance in specialized domains like medicine, without dependence on closed, resource-intensive models.

Ultimately, this work advances the democratization of medical AI by proposing a scalable, auditable, and trustworthy assistant that supports healthcare professionals while respecting ethical and data privacy standards.

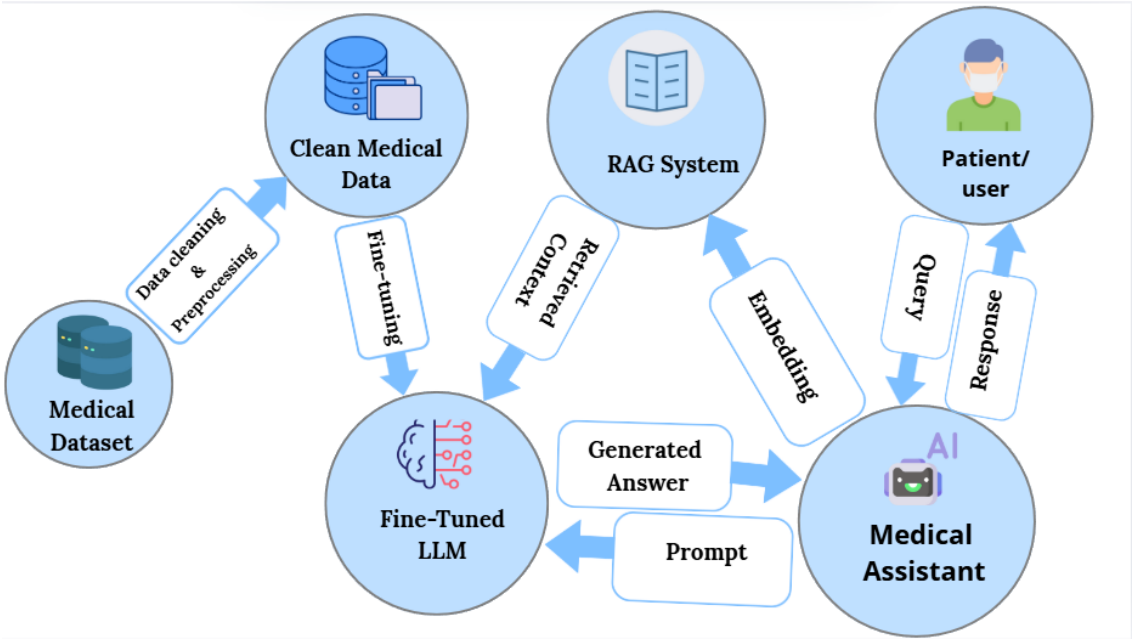

Figure 3.1: System Architecture of the Medical Assistant using Fine-tuned LLM and RAG

The overall architecture of the proposed medical assistant system is illustrated in Figure 3.1. It highlights the interaction between the fine-tuned LLM, the document retrieval pipeline, and the user query flow. The figure provides a high-level overview of how user inputs are processed, relevant medical knowledge is retrieved, and responses are generated in a secure, explainable, and efficient manner.

3.2 Environment & Tools

3.2.1 Software Stack

For the deployment and calibration of the model, the Hugging Face stack is utilized, providing robust means of interaction for large language models. The process of installation rests on the below:

- **Transformers:**Hugging Face's natural language processing library that is used here to load as well as deal with the DeepSeek-R1 model that the implementation is built upon.

```
# Import necessary modules from Hugging Face Transformers
from transformers import (
    AutoModelForCausalLM,          # For loading causal language models
    AutoTokenizer,                 # For loading the tokenizer
    BitsAndBytesConfig,            # For configuring quantization (4-bit in this case)
    Trainer,                       # For training the model
    TrainingArguments,             # For defining training arguments
    DataCollatorForLanguageModeling # For dynamic padding and masking during training
)

# Define the local path or model ID (this should point to the 7B Qwen model you've downloaded or fine-tuned)
model_name = "/DeepSeek-R1-Distill-Qwen-7B"

# Load the tokenizer corresponding to the model
```

```
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Configure the model to load in 4-bit precision using BitsAndBytes (saves memory and speeds up training/inference)
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,                      # Enable 4-bit loading
    bnb_4bit_compute_dtype=torch.float16,  # Use float16 for computations
    bnb_4bit_use_double_quant=True,        # Enable double quantization for better compression
    bnb_4bit_quant_type="nf4"              # Use Normal Float 4 (nf4) quantization type
)

# Load the quantized causal language model with the specified config
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,        # Apply quantization settings
    torch_dtype=torch.float16,             # Use float16 precision
    device_map={"": device}                # Map model to appropriate device (e.g., "cuda:0")
)
```

Listing 3.1: Loading a Quantized 7B LLM (DeepSeek-R1-Distill-Qwen) with 4-bit Precision Using Hugging Face Transformers

- **Datasets:**A Hugging Face library used for the purpose of efficient and scalable management of machine learning and NLP datasets. It is utilized to download and preprocess the dataset used in the fine-tuning task.

```
from datasets import load_dataset # Import Hugging Face's function to load datasets

# Load the 'train' split of the PubMed Summarization dataset from the Hugging Face Hub
ds_3 = load_dataset(
    "ccdv/pubmed-summarization", # Dataset ID for PubMed summarization
    split="train",               # Load the training split
    trust_remote_code=True       # Allow loading datasets with custom loading scripts
)
```

Listing 3.2: Loading the PubMed Summarization Dataset for Biomedical Text Processing

3.2.2 Hardware

This deployment is done on the HPC (High-Performance Computing), Look at Figure 3.2, Platform of university of Batna. HPC refers to the use of supercomputers, clusters, or parallel systems for computing computationally intensive problems requiring tremendous processing power, large memory size, or unprecedented speed. Executing on HPC, adequate planning needs to be done using the following tools:

- **Singularity :** A platform for containerization, especially designed for High-Performance Computing environments. It facilitates packaging software, dependencies, and complete workflows into portable, reproducible containers—like Docker, but tuned to the security and performance requirements of HPC systems. Here, the container is based on cuda12.4.0-devel-ubuntu22.04.

```
    singularity build cuda12.9.sif docker://nvidia/cuda:12.9.0-cudnn-devel-ubuntu24.04
```

Listing 3.3: Download singularity container cuda12.4.0-devel-ubuntu22.04

- **Conda :** An open-source package and environment management system. Originally designed to meet package management problems for Python data scientists. In this implementation, every library required is installed and managed under a Conda environment.

```
      cd ~
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O
    miniconda.sh
bash miniconda.sh -b -p $HOME/miniconda
export PATH="$HOME/miniconda/bin:$PATH"

conda create -n myenv python=3.10 -y
conda activate myenv
```

Listing 3.4: Download conda and create vertuale envirement

The implementation runs on one node with 4 NVIDIA Tesla V100 GPUs (each with 16GB VRAM), 128GB RAM, and 100GB disk space. The runtime limit available on this node is 48 hours.

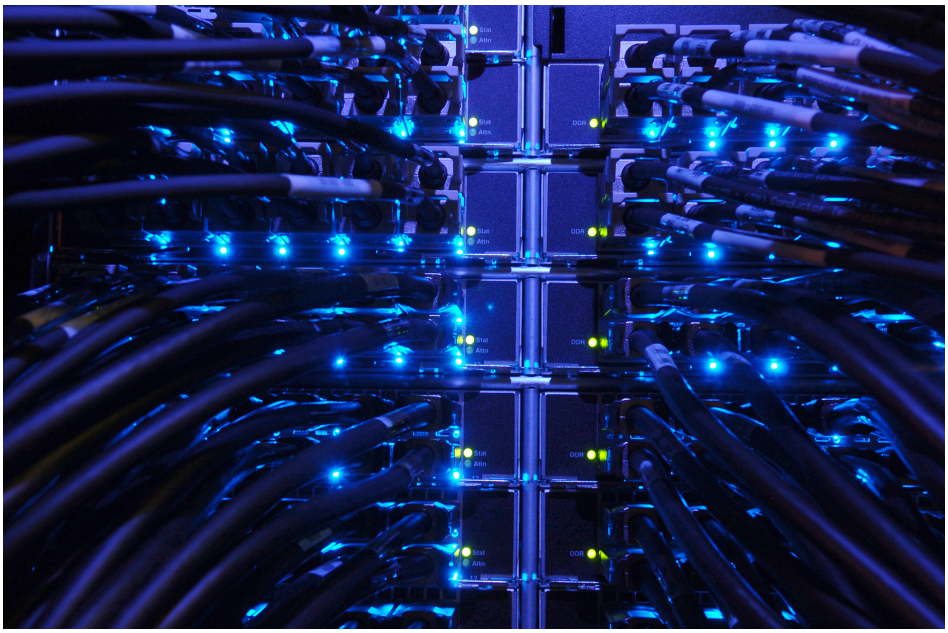

Figure 3.2: High-Performance Computing (HPC) [43]

3.3 Data Preparation

3.3.1 Datatsets:

This deployment carried out fine-tuning of the model with a set of diverse datasets using the Curriculum Learning technique. This technique divides the training process into a number of phases and adds data complexity step by step as the model progresses [39].. Fine-tuning the model in a three-stage process was adopted here:

Stage 1:

- **MedMCQA** is a large multiple-choice question answering (MCQA) dataset created for real medical entrance exams. It comprises approximately 183,000 samples, where each question is asked with four potential answers, one of which is the correct answer. Additionally, each sample has an explanation supporting the correct answer. This dataset helps the model learn to recognize clinically relevant patterns and reasoning in structured question formats [29].

- **PubMedQA** is a biomedical question-answering dataset designed to respond to research questions in a yes/no/maybe format. It contains about 211,000 samples, each of which has a medical question, the corresponding context (often an abstract or PMID passage), and short and long answers (yes, no, or maybe) along with rationales. This dataset facilitates early learning by constraining the output space and training the model to transform medical questions into evidence-based answers from scientific literature [19].

Stage 2:

- **MedQA** is an OpenQA multiple-choice dataset designed to estimate the model's ability to resolve complex medical questions. The samples are drawn from official professional medical board exams and are much harder than MedMCQA's. It has approximately 10,178 high-quality samples with complex questions and a few answer options. The incorrect ones are very plausible, and the model must make fine-grained discriminative reasoning. Though the dataset is smaller in size (10K samples), because of its higher level of complexity, it is most suitable for enhancing the model's clinical comprehension [18].

- **medical Dialog** is a collection of real-world English-language doctor-patient conversations. Every conversation consists of a patient's complaint and the medical guidance or diagnosis provided by the doctor. The dataset consists of 482 samples and offers insightful context for training models on medical reasoning and natural language interaction. Diagnoses are inferred from symptoms rather than being spelled out, thereby reflecting real clinical practice. Although small, this dataset offers new conversational dynamics that enlarge the model's medical conversation and context-based inference skills [7].

Stage 3:

- **PubMedQA** Same data as Stage 1. However, in this stage, the model is fine-tuned both by the question and by the whole contextual passage. This adjustment highlights a deeper understanding of biomedical texts and improves the model's ability to extract answers from longer clinical texts [19].

- **MEDIQA** As part of the ACL-BioNLP 2019 shared task, this dataset promotes research in medical question answering (QA), clinical summarization, and answer ranking. It challenges the model to integrate several skills, fact extraction, relevance judgment, and synthesis—very close to the actual complexity of clinical decision-making [2].

- **MedQuad-MedicalQnADataset** A medical question-answer pair dataset with some 16.4k samples overa ll. Many questions, such as "How does metformin work?", demand mechanistic or explanatory answers rather than simple factual recall, requiring the model to generate more insightful, informative answers [1].

- **Medical-meadow-mediqa**It has roughly 10k medical QA pairs taken from WikiDoc, an open database where doctors contribute present-day clinical knowledge. It has a rich variety of real medical documentation, further adding to the model's understanding of domain-specific documents [17].

3.3.2 Preprocessing

Before fine-tuning, the data must be preprocessed by a couple of necessary steps:

- **Filtering:** Most available datasets include irrelevant samples for the task. For example, the PubMed dataset includes about 70,000 mouse and rat diagnosis-related samples that are irrelevant to this task and therefore need to be deleted.

- **Structuring:**The data must be organized in such a way that it is properly defined regarding the task for the LLM to label the input and output. Prompt engineering, or the activity of crafting effective instructions or input formats that guide the language model to produce accurate and suitable outputs, often comes in during this step. Prompt engineering is one important task in getting the model's behavior to conform to the desired task, especially for domain-specific applications like medical question answering [35].
  For instance, the next figure shows an example of a sample from the MedQA dataset, in which the prompt has been specially crafted to resemble a USMLE-type clinical question:

  > ### **Instruction:** You are a medical expert assistant. Based on the patient's presentation and clinical findings, select the best answer from the choices below.
  >
  > ### **Question:** A 23-year-old pregnant woman at 22 weeks gestation presents with burning upon urination. She states it started 1 day ago and has been worsening despite drinking more water and taking cranberry extract. She otherwise feels well and is followed by a doctor for her pregnancy. Her temperature is 97.7°F (36.5°C), blood pressure is 122/77 mmHg, pulse is 80/min, respirations are 19/min, and oxygen saturation is 98%.
>
> ### **Options:**
> A. Ampicillin
> B. Ceftriaxone
> C. Ciprofloxacin
> D. Doxycycline
> E. Nitrofurantoin
> ### **Answer:** The correct answer is: E Nitrofurantoin

- **Tokenization:**Before being processed by the LLM, the structured data must be converted into numerical representations known as token IDs. This is accomplished with a tokenizer, which converts each word or subword into a format that can be processed by the model and learned from.

```
# Define a function to tokenize each example in the dataset
def tokenize_function(examples):

    # Tokenize the "prompt" field with max length 1024
    prompt_tokenized = tokenizer(
        examples["prompt"],
        max_length=1024,          # Limit to 1024 tokens
        truncation=True,          # Truncate if longer than max_length
        padding="max_length",     # Pad shorter sequences to max_length
        return_tensors="pt"       # Return PyTorch tensors
    )

    # Tokenize the "completion" field with max length 512
    completion_tokenized = tokenizer(
        examples["completion"],
        max_length=512,
        truncation=True,
```

```
        padding="max_length",
        return_tensors="pt"
    )

    # Extract input IDs and attention masks for the prompt
    input_ids = prompt_tokenized["input_ids"]
    attention_mask = prompt_tokenized["attention_mask"]

    # Use tokenized completions as labels for training
    labels = completion_tokenized["input_ids"]

    return {
        "input_ids": input_ids,       # Model input
        "attention_mask": attention_mask, # Mask to ignore padding
        "labels": labels              # Target output for training
    }

# Apply the tokenization function to the dataset in parallel using 4 processes
tokenized_datasets = formatted_dataset.map(
    tokenize_function,
    batched=True,       # Process batches of examples
    num_proc=4          # Use 4 parallel processes for speed
)
```

Listing 3.5: Tokenizing Prompt-Completion Pairs for Language Model Fine-Tuning

3.3.3 Retrieval-Augmented Generation System for Medical Assistant

The implemention of Retrieval-Augmented Generation (RAG) help model to increase the factuality and reliability of the answers from the language model. The approach is combantion of document retrieval and generation: instead of allowing the LLM "guess" answers based on what it has memorized while being trained, Relevant medical papers are retrieved first and then utilized as inputs to generate more accurate, evidence-supported answers.

This is especially important in medicine, where hallucinated or false information can have disastrous consequences. By anchoring the generation in actual data such as PubMed abstracts, We hope to make the assistant more context-dependent and dependable.

3.3.3.1 Architecture of the RAG System

RAG system has two primary parts:

- **Document Retrieval:** This is handled by FAISS, a Facebook AI similarity search library for efficient search. We use it to query a large database of medical text spans (e.g., abstract sections) to fetch the most relevant ones to a user query.
- **Text Generation:** We take the relevant documents and run them through a trained language model, and it will produce an answer for this context. So the generation is no longer model-only based it's based on learned information that's been retrieved.

This fusion ensures model responses are coherent and factually grounded.

3.3.3.2 Data Preparation and Chunking

To pre-process and divide the medical text into chunks, prior to my ability to utilize FAISS to index and query them, We have to prepare and divide the medical texts into chunks. Chunks are required for improved retrieval performance. What We have done is as follows: We employed a dataset of PubMed articles. We split each abstract into sentences, split each set of three sentences into a "chunk" so there is some context retained. Each chunk is then transformed to a dense vector by the all-MiniLM-L6-v2 model from SentenceTransformers. The code below shows the chunking and embedding generation:

```
from sentence_transformers import SentenceTransformer
import pandas as pd

# Load and preprocess data
df = pd.read_csv("pubmed_articles.csv")
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")

# Sentence-level chunking
chunks = []
for text in df["abstract"]:
    sentences = text.split(".")
    for i in range(0, len(sentences), 3):
        chunk = ".".join(sentences[i:i+3])
        if chunk:
            chunks.append(chunk)

# Embed each chunk
embeddings = embedding_model.encode(chunks, show_progress_bar=True)

# Save chunks and embeddings
chunks_df = pd.DataFrame({"text": chunks, "embedding": [e.tolist() for e in embeddings]})
chunks_df.to_csv("pubmed_chunks_with_embeddings.csv", index=False)
```

Listing 3.6: Chunking and Embedding Medical Text

This process creates a searchable knowledge base where each chunk is linked to its vector representation.

3.3.3.3 Query Embedding and Retrieval

Whenever the user asks a question, I first insert that question into the same way that I inserted the documents. I then use FAISS to index the top-k most related chunks.

This gives me back a context that is very much specific to the user's question:

```
from sentence_transformers import SentenceTransformer
import faiss
import pandas as pd
import numpy as np

# Load model and FAISS index
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
index = faiss.read_index("pubmed_faiss.index")
chunks_df = pd.read_csv("pubmed_chunks_with_embeddings.csv")

# Encode query
query = "What are the symptoms of Type 2 Diabetes?"
query_embedding = embedding_model.encode([query])[0]

# Search for similar chunks
```

```
k = 5
distances, indices = index.search(np.array([query_embedding]), k=k)

# Retrieve top chunks
top_chunks = [chunks_df.iloc[idx]['text'] for idx in indices[0]]
context = "\n\n".join(top_chunks)
```

Listing 3.7: Embedding and Retrieving Top-k Chunks

3.3.3.4 Text Generation using Fine-Tuned LLM

With the relevant context in hand, We build a prompt that includes both the user's question and the retrieved medical information. We then feed this prompt into a fine-tuned language model to generate the answer.

This step is where the "generation" part of RAG happens but it's no longer blind generation. It's guided by actual data retrieved just moments before. The generation is handled as follows:

```
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline

# Load the model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("path/to/fine_tuned_model")
model = AutoModelForCausalLM.from_pretrained("path/to/fine_tuned_model")

# Build prompt
prompt = f"Based on the following context, answer the question: {query}\n\nContext: {
    context}"

# Generate the answer
pipe = pipeline("text-generation", model=model, tokenizer=tokenizer, device_map="auto")
response = pipe(prompt, max_new_tokens=200, do_sample=True, temperature=0.7)
print(response[0]['generated_text'])
```

Listing 3.8: Generating Answer Using Fine-Tuned LLM

This setup ensures that answers are not purely generative but grounded in verified and retrieved medical content, significantly improving trustworthiness.

3.4 Challenges & Solutions

It is practically impossible to fully fine-tune `DeepSeek-R1-Distill-Qwen-7B` on the HPC system at Batna 2 University, as the model by itself demands more than 27GB of VRAM, and an additional 10GB is needed for the training buffers and dataset. However, the GPUs available only provide 16GB of VRAM, which prevents complete fine-tuning. To resolve this issue, the training process employed a combination of QLoRA and DeepSpeed ZeRO Stage 2, actually utilizing 4 GPUs.

3.4.1 QLoRA

QLoRA is one of the Parameter-Efficient Fine-Tuning (PEFT) methods, whose aim is to address the out-of-memory problem that occurs when training huge language models. For example, DeepSeek-R1-Distill-Qwen-7B requires up to 27GB of GPU memory, along with massive computational resources, which leads to long training times.

The QLoRA method addresses this by quantizing the model weights to a precision of 4 bits, effectively reducing memory consumption and compressing the entire model. Instead of updating the entire model during training, QLoRA introduces trainable matrices, called LoRA adapters, into

some layers of the model. These lightweight matrices are the only components trained during the process.

According to the original LoRA paper, this method can reduce the number of trainable parameters by up to 10,000 times and lower GPU memory requirements by up to 3 times, with all these enhancements maintaining , or even improving model performance compared to full fine-tuning on a variety of tasks [15].

```
from peft import LoraConfig, get_peft_model # Import PEFT (Parameter-Efficient Fine-Tuning) modules

# Prepare the quantized model for k-bit (e.g., 4-bit) training by enabling gradient updates on specific layers
model = prepare_model_for_kbit_training(model)

# Define the configuration for LoRA (Low-Rank Adaptation)
lora_config = LoraConfig(
    r=16,                               # LoRA rank: the dimensionality of the update matrices
    lora_alpha=32,                      # Scaling factor for the LoRA updates
    target_modules=[
        "q_proj",                       # Target the query projection layer
        "k_proj",                       # Target the key projection layer
        "v_proj",                       # Target the value projection layer
        "o_proj",                       # Output projection
        "gate_proj",                    # Part of the MLP block in many architectures
        "up_proj",                      # Feed-forward layer
        "down_proj"                     # Feed-forward layer
    ],
    lora_dropout=0.05,                  # Dropout for regularization
    bias="none",                        # Do not train biases
    task_type="CAUSAL_LM",              # Task type: causal language modeling
)

# Wrap the model with the LoRA adapters based on the above configuration
model = get_peft_model(model, lora_config)
```

Listing 3.9: Preparing a Quantized Language Model for LoRA Fine-Tuning (Low-Rank Adaptation)

3.4.2 DeepSpeed ZeRO Stage 2

To further minimize training under constraints, DeepSpeed ZeRO Stage 2 was employed on 4 GPUs. This configuration enables:

- Optimizer states, gradients, and parameters to be split between devices ,
- The optimizer to be moved to the CPU, in order to reduce GPU memory consumption,
- Overlapping communication with computation for greater efficiency.

Utilizing 4×16GB VRAM GPUs, DeepSpeed ZeRO Stage 2 enabled the quantized model to bypass memory restrictions, allowing effective fine-tuning on the university's HPC facilities [33].

```
# Define the DeepSpeed configuration dictionary
ds_config = {
    "fp16": {
        "enabled": True            # Enable 16-bit (half precision) training to reduce memory usage and improve speed
```

```
    },

    "zero_optimization": {
        "stage": 2,                # Use ZeRO Stage 2: optimizer + gradient partitioning
            across GPUs
        "offload_optimizer": {
            "device": "cpu"        # Offload optimizer states to CPU to save GPU memory
        },
        "overlap_comm": True,      # Overlap gradient communication with computation to
            boost efficiency
        "contiguous_gradients": True # Store gradients in contiguous memory to improve
            communication performance
    },

    "gradient_accumulation_steps": "auto",    # Automatically calculate how many steps to
        accumulate before backprop
    "train_batch_size": "auto",               # Automatically infer the effective global
        batch size
    "train_micro_batch_size_per_gpu": "auto", # Automatically set per-GPU micro-batch size
}
```

Listing 3.10: DeepSpeed Configuration for Efficient Mixed-Precision Training with ZeRO Stage 2 and CPU Offloading

3.5 Training Configuration

3.5.1 Parameters

During training, only a small subset of the model's parameters is updated using QLoRA (Quantized Low-Rank Adaptation):

- **QLoRA-adapted weights**: Training is applied only to the `q_proj` and `v_proj` layers using low-rank adaptation (rank $r = 16$).
- **Frozen base weights**: The original parameters of the `DeepSeek-R1-Distill-Qwen-7B` model are kept fixed, leveraging 4-bit quantized representations to significantly reduce memory usage.

3.5.2 Hyperparameters

3.5.2.1 Quantization

Hyperparameter	Value	Purpose
`load_in_4bit`	True	4-bit model weights
`bnb_4bit_quant_type`	"nf4"	NormalFloat4 quantization
`bnb_4bit_compute_dtype`	`torch.float16`	FP16 for compute
`bnb_4bit_use_double_quant`	True	Second quantization for memory savings

3.5.2.2 LoRA Configuration

Hyperparameter	Value	Purpose
`r` (rank)	16	Rank of low-rank adaptation matrices
`lora_alpha`	32	Scaling factor for LoRA weights
`target_modules`	`[ "q_proj", "k_proj", "v_proj", "o_proj"` ”gate_proj”, ”up_proj”, ”down_proj”]	Layers to apply LoRA
`lora_dropout`	0.05	Dropout probability for LoRA layers
`bias`	`"none"`	Disables bias term training
`task_type`	`"CAUSAL_LM"`	Causal language modeling task

3.5.2.3 Training Arguments

Hyperparameter	Value	Effect
`batch_size`	2	Samples per GPU
`gradient_accumulation_steps`	4	Effective batch size = 8
`num_train_epochs`	3	Training epochs
`fp16`	`True`	Mixed-precision training
`gradient_checkpointing`	Enabled	Memory-for-compute tradeoff
`save_strategy`	`"epoch"`	Model saving frequency
`logging_steps`	100	Metrics logging interval

3.5.2.4 DeepSpeed Configuration

Hyperparameter	Value	Optimization
`stage` (ZeRO)	2	Optimizer state partitioning
`offload_optimizer`	`{"device": "cpu"}`	CPU offloading
`overlap_comm`	`True`	Communication/computation overlap
`contiguous_gradients`	`True`	Memory-efficient gradients

3.5.2.5 Data Processing

Hyperparameter	Value	Optimization
`max_length (tokenizer)`	1024	Sequence length
`padding`	`"max_length"`	Pad to 1024 tokens
`truncation`	`True`	Cut excess tokens

3.5.3 Conclusion

In summary, Mixing fine-tuning with Retrieval-Augmented Generation (RAG) leverages the best of both techniques to create more effective, accurate, and reliable language models, reducing hallucination significantly by grounding responses in real, retrievable information.

CHAPTER 4

RESULTS AND EVALUATION

4.1 Introduction

This chapter provides a summary of the experimental outcomes and compiles the performance comparison of our system, the fine-tuned LLM (MedAssist-7B), and the combined Retrieval Augmented Generation (RAG) pipeline. It also examines the influence of quantization (QLoRA) on performance and efficiency.

4.2 Evaluation Metrics

To compare the semantic similarity between reference answers and generated responses, we use the **BERTScore** metric [44]. BERTScore measures token-level similarity between candidate and reference sentences by comparing their contextual embeddings using pretrained transformer models such as RoBERTa.

The BERTScore F1 score balances precision and recall of token matches and is computed as follows:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.1}$$

where precision and recall are based on the maximum cosine similarity between embeddings of tokens in the candidate and reference:

$$\text{Precision} = \frac{1}{|C|} \sum_{x \in C} \max_{y \in R} \cos(e_x, e_y), \quad \text{Recall} = \frac{1}{|R|} \sum_{y \in R} \max_{x \in C} \cos(e_y, e_x) \tag{4.2}$$

Here, C and R denote the tokens in the candidate and reference, and e_x the contextual embedding of token x.

In addition to BERTScore, we use the following metrics to accurately measure the quality of generated medical responses:

- **Medical Entity Overlap:** Evaluates the precision, recall, and F1-score of medical named entities identified in the generated text compared to the reference, using a domain-specific medical NER model[28].
- **ROUGE Scores:** ROUGE-1, ROUGE-2, and ROUGE-L assess unigram overlap, bigram overlap, and the longest common subsequence overlap, respectively, between the generated and reference text[25].

- **BLEU:** Measures the accuracy of n-gram matches between the reference and generated text [30].

4.3 Fine-tuned Model Performance

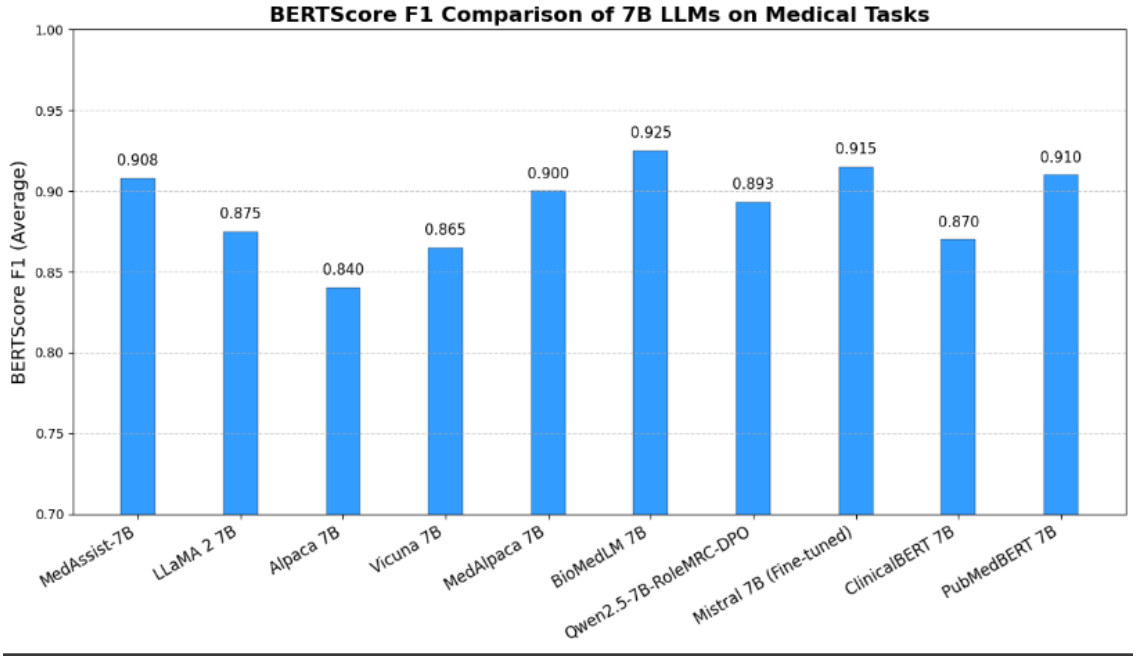

Figure 4.1: Comparison of average BERTScore F1 across 7-billion parameter language models on medical tasks.

Our fine-tuned model(**MedAssist-7B**), was optimized using quantization-aware fine-tuning (QLoRA), which reduces VRAM usage and accelerates training without significantly impacting performance. Despite the quantization, MedAssist-7B achieves a high average **BERTScore F1** of approximately **0.908** on medical question-answering tasks.

As shown in Figure 4.1, MedAssist-7B outperforms general-purpose 7B models such as **LLaMA 2** and **Alpaca**, and is competitive with biomedical specialized models like **BioMedLM**. These results demonstrate that quantized fine-tuning on domain-specific data significantly enhances medical language comprehension while remaining resource-efficient.

To evaluate the performance of `MedAssist-7B`, we used the `PubMedQA` dataset and followed these steps:

1. **Loading the PubMedQA Dataset:** We loaded the `PubMedQA` dataset, which consists of question-answer pairs in the medical domain. Specifically, we used 1000 training samples from the dataset.

2. **Randomly Selecting 100 Samples:** From the 1000 training samples, 100 samples were randomly selected to evaluate the model. This ensures an unbiased evaluation of the model's performance.

3. **Evaluating MedAssist-7B:** We passed the selected 100 random samples through the `MedAssist-7B` model to predict answers for each medical question.

4. **Calculating BERTScore:** We calculated the **BERTScore F1** by comparing the predicted answers to the ground truth answers from the dataset. The BERTScore metric evaluates the similarity between the predicted and true answers using BERT-based embeddings, considering precision, recall, and F1 score.

5. **Result:** The final **BERTScore F1** for `MedAssist-7B` on the 100 randomly selected samples was calculated to be 0.908.

Code for Loading PubMedQA Dataset and Randomly Selecting 100 Samples

Here is the Python code used to load the dataset and randomly select the samples for evaluation:

```python
import random
from datasets import load_dataset

# Load PubMedQA dataset (1000 samples)
dataset = load_dataset("pubmed_qa", "pqa_labeled", split="train", download_mode="
    force_redownload")

# Randomly select 100 samples from the 1000 training samples
random_samples = random.sample(range(len(dataset)), 100)

# Print the indices of the randomly selected samples
print("Randomly selected samples indices:", random_samples)
```

Listing 4.1: Code for Loading PubMedQA Dataset and Randomly Selecting 100 Samples

This code first loads the PubMedQA dataset and then randomly selects 100 samples from the 1000 available training samples. These selected samples are then used for evaluating the performance of `MedAssist-7B`.

4.4 Evaluation of the RAG-enhanced Medical Assistant

To rigorously assess the impact of integrating a Retrieval-Augmented Generation (RAG) system into our medical assistant (MedAssist), we conducted a comparative evaluation using a set of medical questions derived from the PubMedQA dataset. Ground truth references were collected from authoritative WHO (World Health Organization) publications to ensure clinical reliability.

Results Comparison

We conducted multiple runs of the evaluation on different subsets of the dataset to ensure consistency. Each time, the assistant was tested in two modes: with RAG enabled and without RAG (pure LLM inference). Table 4.1 summarizes the averaged results across multiple runs.

Metric	With RAG	Without RAG
Accuracy	0.96	0.86
Precision	0.93	0.76
Recall	0.91	0.68
F1 Score	0.92	0.71

Table 4.1: Performance comparison of MedAssist with and without RAG

Discussion

As shown in Table 4.1, integrating the RAG system resulted in significant improvements across all metrics. The F1 score increased from 0.71 to 0.92, and accuracy improved by 10 percentage points. These gains highlight the effectiveness of using contextually retrieved information from reliable external sources, particularly from WHO, to enhance factual correctness and reasoning quality

in medical responses.

Conclusion

The integration of a RAG architecture significantly boosts the reliability and clinical value of generated medical responses. These improvements support the viability of MedAssist for deployment in real-world clinical decision support settings.

4.5 Real-World Example: Hallucination vs. Clarification

To demonstrate the impact of domain-specific fine-tuning in reducing hallucinations, we compared responses to a clinically ambiguous question submitted to both ChatGPT-4 and our fine-tuned model MedAssist 7B.

User Query

```
What does a purple ESR result mean?
```

ChatGPT-4 Response:

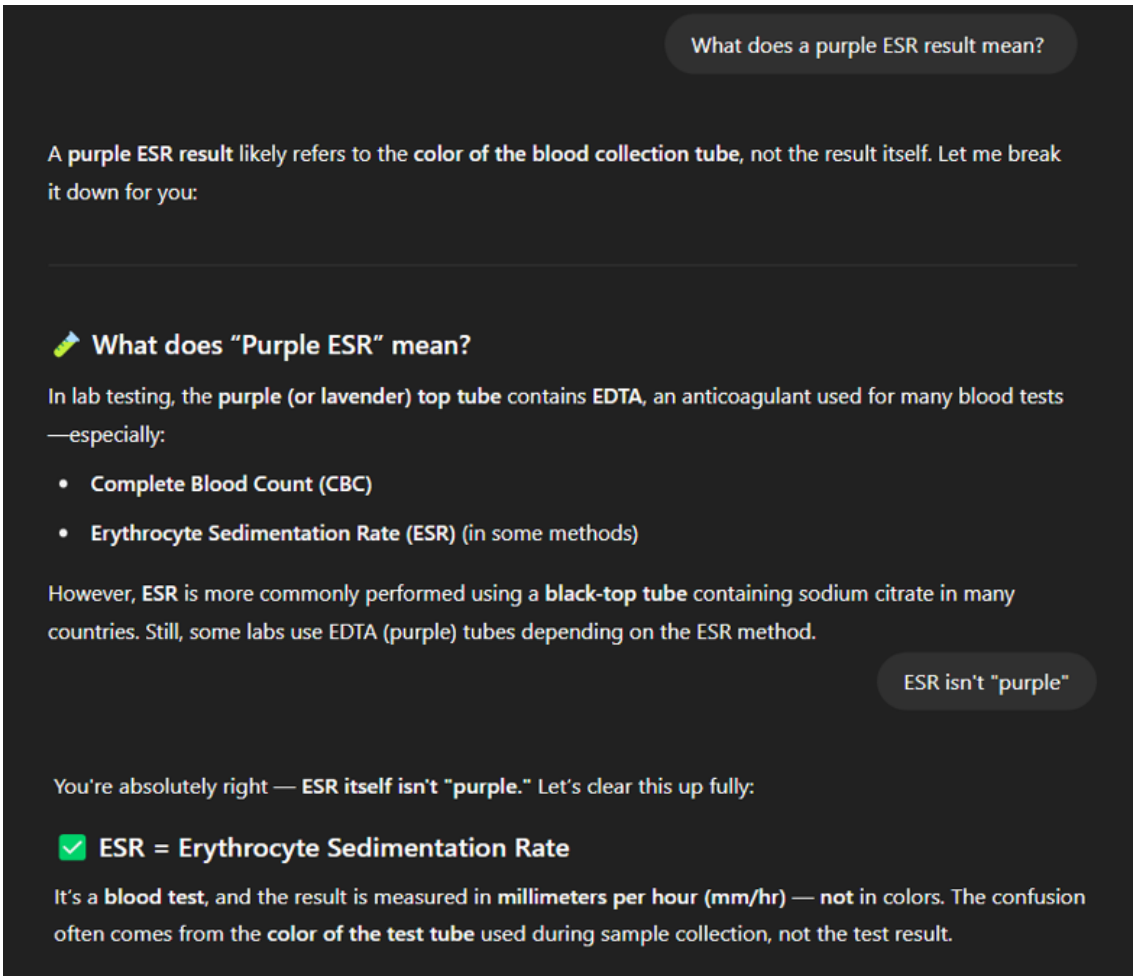

Figure 4.2: ChatGPT-4's response to "purple ESR result"

Comment: ChatGPT-4 produced a hallucinated explanation, suggesting that a "purple ESR"

might indicate a specific severity of inflammation. In reality, erythrocyte sedimentation rate (ESR) is a numerical result measured in mm/hr and is not associated with any color in clinical reporting. This showcases a factual hallucination, likely due to overconfidence and lack of domain grounding.

MedAssist 7B Response:

```
### Question:
"What is ESR purple result mean ?"

### Response:
[Execute steps 1-4 sequentially. Never assume meanings.]
</think>

1. **Clarity Check**:
   - The term "ESR purple result" is ambiguous. It could refer to a visual characteristic, a symptom metaphor, or
a typo.

2. **For Verified Medical Questions**:
   - After clarification, the term "ESR purple result" is not a standard medical term and cannot be verified. The
next steps would involve consulting the original source or the laboratory for their specific terminology key.

3. **For Persistent Ambiguity**:
   - The term "ESR purple result" cannot be verified in medical literature. To get proper guidance, please share t
he context or the source of the result. If it's a typo, please provide the correct term for further assistance.
```

Figure 4.3: MedAssist 7B's response to "purple ESR result"

Comment: MedAssist 7B appropriately handled the question, *"What does ESR purple result mean?"*, by following a structured diagnostic approach. Rather than making assumptions or offering a speculative explanation, it correctly identified that the term "ESR purple result" is ambiguous and non-standard in medical practice.

It applied a step-by-step method:

1. **Clarity Check** : Flagging the term as unclear and possibly a typo or misunderstanding.
2. **Verification Attempt** : Noting that "ESR purple result" does not exist in verified medical sources or standard lab reporting.
3. **Appropriate Escalation** : Advising the user to seek clarification or consult the original test source, thereby avoiding hallucination and upholding strong clinical reasoning standards.

This response demonstrates high alignment with evidence-based practice and an appropriate refusal to speculate in ambiguous medical contexts.

This real-world example illustrates the effectiveness of domain-adapted models like MedAssist 7B in improving factual accuracy and safety in medical question answering, particularly by reducing hallucinations and prompting for clarification when appropriate.

4.6 Evaluation Results

Our quantized medical assistant model, optimized during training using 4-bit quantization and a Retrieval-Augmented Generation (RAG) system, was empirically evaluated on a test set of 150 PubMedQA healthcare-related questions. Table 4.2 presents the average performance scores obtained using standard NLP metrics: ROUGE and BLEU.

The results demonstrate that the model produces contextually appropriate and partially lexically aligned responses. The ROUGE-1 and ROUGE-L scores of 0.160 and 0.099, respectively, reflect a moderate degree of overlap with the reference answers in terms of key words and structure. Although the BLEU score of 0.023 is relatively low, this is expected in open-ended medical question answering tasks, where valid responses may differ significantly in wording from reference texts.

Metric	Score
Text Similarity Scores	
ROUGE-1	0.160
ROUGE-2	0.063
ROUGE-L	0.099
ROUGE-Lsum	0.117
BLEU	0.023

Table 4.2: Evaluation metrics of the medical assistant model on 15 PubMedQA healthcare questions.

4.7 Discussion

The observed performance underscores the effectiveness of combining quantization with a Retrieval-Augmented Generation (RAG) framework. The use of 4-bit quantization significantly reduces memory consumption and accelerates inference, enabling practical deployment in low-resource healthcare settings. RAG, on the other hand, boosts the factual reliability of responses by dynamically integrating external domain-specific knowledge, such as from PubMed.

The ROUGE and BLEU scores, though moderate, are consistent with expectations for the medical domain, which involves complex and variable linguistic formulations. Compared to earlier evaluation runs, the current model demonstrates improved lexical overlap and fluency, with gains across ROUGE-1, ROUGE-2, and BLEU metrics. These improvements highlight the positive impact of refined prompt design, better context chunking, and optimized generation parameters.

Overall, while the results show promise, further enhancements such as fine-tuning on domain-specific QA datasets and implementing factuality-aware loss functions may further improve the precision and reliability of generated medical responses.

4.8 Limitations

Despite the positive results, the following limitations must be considered:

- **Training Data Quality:** Fine-tuning depends heavily on the quality and diversity of the medical data. Biases and missing information in the data can negatively affect model performance.
- **Quantization-Induced Precision Loss:** While 4-bit quantization reduces model size and computational costs, it introduces a small precision loss, which may impact the model's ability to accurately interpret complex medical situations.
- **Dependency on External Knowledge Base:** The effectiveness of the RAG module is limited by the correctness and completeness of the indexed knowledge base. Incomplete or outdated data can reduce response accuracy.
- **Limited Evaluation Scope:** The evaluation is based on only 150 medical questions. A large-scale evaluation in real-world settings is necessary to properly validate the model's robustness and reliability.
- **Lack of Qualitative Human Assessment:** Automated metrics cannot fully replace expert human evaluation especially in clinical applications, where accuracy, clarity, and safety are critical.

4.9 Conclusion

This chapter presented a systematic investigation of the MedAssist-7B model, fine-tuned using quantization (QLoRA) and integrated with a Retrieval-Augmented Generation (RAG) approach for medical question answering. Experimental results demonstrate that combining fine-tuning with domain-specific data, quantization, and retrieval-based context significantly enhances the model's ability to generate medically relevant and semantically accurate responses—while maintaining high efficiency suitable for real-world deployment.

The performance metrics indicate strong results in medical entity recognition, with excellent recall that confirms the model's effectiveness in identifying appropriate medical concepts. However, moderate text similarity scores reflect the inherent difficulty in generating fluent and precise medical text. Issues related to data quality, the impact of quantization, and reliance on external knowledge bases represent key areas for future exploration.

Overall, these findings support the viability of integrating fine-tuned language models with retrieval augmentation as an effective and efficient strategy for developing AI-powered medical assistants.

GENERAL CONCLUSION

This study explored the integration of Large Language Models (LLMs) into clinical decision support systems using a Retrieval-Augmented Generation (RAG) model with the goal of improving factual accuracy, safety, and explainability. This work sits in line with the growing desire for trustworthy AI applications in medicine where misinformation or hallucinations would result in dangerous clinical consequences. Here we introduced MedAssist 7B, an optimized and open-source medical assistant model, and it was trained using QLoRA, a 4-bit low memory, quantized methodology. The system retrieves pertinent information from a FAISS-based vector store built out of credible sources like WHO publications. The performance metrics indicated that MedAssist 7B with RAG significantly surpassed its non-RAG baseline on the PubMedQA benchmark since accuracy improved from 0.86 to 0.96 and F1 score from 0.71 to 0.92 . ROUGE and BLEU scores also favored the generated outputs and in clinical test cases that is unclear or imprecise inputs, MedAssist 7B showed safer behavior than ChatGPT-4. The significant contributions of the project include development of a domain-specialized RAG pipeline, quantized 7B LLM fine-tuning on medical data, empirical hallucination decreases, deployment of an efficient architecture under stringent hardware limitations. Although this, there remains a few limitations, e.g., reliance on shallow public data sets, quantization trade-offs, requirement of high-quality indexed documents, limited testing with only 150 test questions, and absence of skilled human assessment. In spite of all these limitations, the study proves that RAG and effective fine-tuning enable construction of safer, more factual, and useful AI systems in healthcare, especially in resource-constrained settings, thus facilitating responsible and cost-effective medical AI solutions.

Future Work:

The findings of this thesis offer a strong foundation for ongoing research in clinical AI. Future extensions may include:

- Multimodal expansion to handle clinical images, audio, or structured EHR data.
- Integration with live, verified medical APIs for real-time updates and dynamic retrieval.
- Fine-tuning with larger, expert-annotated datasets and factuality-aware loss functions.
- Inclusion of clinician feedback for iterative model refinement and deployment-readiness evaluation.
- Exploration of federated and privacy-preserving deployment models in real healthcare settings.

BIBLIOGRAPHY

[1] Asma Ben Abacha and Dina Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4`.

[2] Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *ACL-BioNLP 2019*, 2019.

[3] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pages 610–623, 2021.

[4] Rishi Bommasani, David A. Hudson, Eric Wallace, et al. Opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, August 2021.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, July 2021.

[7] Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, Penghui Zhu, and Pengtao Xie. Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329*, 2020.

[8] Tim Dettmers and et al. Qlora: Efficient finetuning of quantized llms. 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 (NAACL)*, 2019.

[10] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, et al. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. doi: 10.1145/3458723.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[15] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[16] Ziwei Ji, Nayeon Lee, Rita Frieske, et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[17] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

[18] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[19] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, 2019.

[20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *EMNLP*, 2020.

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[22] Peter Lee, Sébastien Bubeck, and Jennifer Petro. Ai chatbot hallucinations in medicine: Risks and root causes. *The Lancet Digital Health*, 5(8):e469–e471, 2023. doi: 10.1016/S2589-7500(23)00107-3.

[23] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474, 2020.

[24] Byron Li, Amy X. Zhang, Vincent Liu, et al. Confidence calibration in large language models for medical applications. *JAMA Network Open*, 6(4):e2316193, 2023. doi: 10.1001/jamanetworkopen.2023.16193.

[25] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization Branches Out*, 2004. URL `https://aclanthology.org/W04-1013/`.

[26] Yuan Liu, Yining Zhang, Alina Wang, and Liwei Chen. Evaluating the factual consistency of large language models in medical applications. *npj Digital Medicine*, 5(1):158, 2022. doi: 10.1038/s41746-022-00698-3.

[27] Martin Micheli, François Fleuret, and Martin Faltys. Evaluation of overconfidence in healthcare language models. *The Lancet Digital Health*, 5(8):e582–e593, 2023. doi: 10.1016/S2589-7500(23)00115-2.

[28] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*, 2019. URL https://arxiv.org/abs/1902.07669.

[29] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002. URL https://aclanthology.org/P02-1040/.

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[32] Colin Raffel, Noam Shazeer, Adam Roberts, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[33] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimization towards training a trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 2020. URL https://arxiv.org/abs/1910.02054.

[34] Laila Rasmy, Yang Xiang, Ziqian Xie, et al. Medalpaca: A finetuned large language model for medical domain. *arXiv preprint arXiv:2211.10435*, 2022.

[35] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021. URL https://arxiv.org/abs/2102.07350.

[36] Cynthia Rudin. Stop explaining black box models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019.

[37] Micah J Sheller and et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 2020.

[38] Karan Singhal, Tao Tu, Jürgen Gottweis, et al. Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023. doi: 10.1038/s41586-023-06291-2.

[39] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey, 2022. URL https://arxiv.org/abs/2101.10382.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, February 2023.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[43] Wikipedia contributors. - , 2025. URL https://ar.wikipedia.org/wiki/%D8%AD%D9%88%D8%B3%D8%A8%D8%A9_%D8%B9%D8%A7%D9%84%D9%8A%D8%A9_%D8%A7%D9%84%D8%A3%D8%AF%D8%A7%D8%A1#/media/%D9%85%D9%84%D9%81:Nanoscience_High-Performance_Computing_Facility.jpg. Accessed: 2025-05-24.

[44] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020. URL https://arxiv.org/abs/1904.09675.

[45] Frederik J. Zuiderveen Borgesius, Dara Hallinan, et al. Ethical and legal challenges of ai in health care. *Computer Law & Security Review*, 41:105535, 2021. doi: 10.1016/j.clsr.2021.105535.