FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Leveraging Local LLMs in Medtech: Smaller, Effective and Private Gen AI for healthcare solutions

**Ana Luísa Ferreira Marques**


Mestrado em Engenharia Informática

Supervisor: Prof. Gil Gonçalves

February 25, 2025

# Leveraging Local LLMs in Medtech: Smaller, Effective and Private Gen AI for healthcare solutions

**Ana Luísa Ferreira Marques**

Mestrado em Engenharia Informática

Approved in oral examination by the committee:

President: Prof. Rui Camacho
External Examiner: Prof. Fátima Manuela da Silva Leal
Supervisor: Prof. Gil Manuel Gonçalves

February 25, 2025

# Resumo

O uso crescente de LLMs em setores críticos como a saúde ressalta a necessidade de soluções que equilibrem desempenho, custo, consumo de energia e privacidade dos dados. Modelos baseados em serviços, como o GPT-4-1106-preview, demonstram desempenho de ponta, alcançando 32% de precisão em tarefas de resposta aberta a perguntas médicas. No entanto, estes modelos apresentam custos operacionais elevados, consumindo 13.314,4 Joules por resposta e excedendo $1.000.000 anuais em cenários de uso intensivo. Diante deste panorama, esta dissertação investiga se LLMs menores, implementados localmente e aprimorados através da incorporação de RAG, podem obter resultados relevantes no domínio médico comparáveis aos de seus equivalentes baseados em serviços.

A metodologia proposta utiliza conjuntos de dados médicos cuidadosamente selecionados, integrados num pipeline de recuperação de informações que complementa a geração de modelos locais, reduzindo a dependência de arquiteturas de grande escala. Phi4, um LLM local com 14 biliões de parâmetros, foram avaliados em comparação com LLMs de propósito geral e especializados em medicina, analisando o seu desempenho, eficiência energética e custos monetários em benchmarks específicos da área da saúde. O modelo Phi4 alcançou a maior precisão, com 33%, consumindo 86,08 Joules por resposta, demonstrando a sua capacidade de rivalizar com o GPT-4-1106-preview em termos de desempenho, com custos energéticos e operacionais significativamente mais baixos. Modelos locais de médio porte, como o Gemma2-9B, demonstraram equilíbrio entre eficiência energética (68,24 Joules por resposta) e desempenho ligeiramente inferior. Por outro lado, modelos locais menores, como o Llama3.2-3b, apresentaram uma eficiência energética excepcional (2,26 Joules por resposta), mas não atingiram a precisão necessária para tarefas críticas no setor de saúde.

O estudo também destaca os desafios no uso de LLMs especializados em medicina, como o Meditron 7B e o MedLLaMA 2, num framework RAG. O Meditron 7B falhou em gerar respostas consistentes, enquanto o MedLLaMA 2 não conseguiu seguir as instruções do prompt, tornando-os inviáveis para avaliação.

Conclui-se que a integração de LLMs menores e locais, combinados com uma camada dedicada de recuperação de informações, constitui uma estratégia viável para fornecer soluções robustas, contextualmente precisas e eficientes em termos de recursos na área da saúde. Embora aspectos como protocolos de fine-tuning, otimização de parâmetros de recuperação e eficiência energética exijam investigação adicional, os resultados demonstram que LLMs locais, como o Phi4, podem efetivamente reduzir a distância de desempenho em relação a APIs comerciais, oferecendo soluções sustentáveis, economicamente viáveis e centradas na privacidade, alinhadas às rigorosas exigências do setor de saúde.

# Abstract

The increasing prevalence of LLMs in critical sectors such as healthcare underscores a pressing need for solutions that balance performance, cost-efficiency, energy consumption, and privacy. Service-based LLMs, such as GPT-4-1106-preview, consistently demonstrate advanced performance, achieving 32% accuracy in open-answer medical tasks. However, they incur significant monetary and energy costs, consuming 13,314.4 Joules per response and exceeding $1,000,000 annually in high-volume deployments. Against this backdrop, this dissertation investigates whether smaller, locally deployed LLMs, augmented through a Retrieval-Augmented Generation framework, can deliver domain-relevant outputs comparable to their larger, service-based counterparts.

In the proposed methodology, curated medical datasets are embedded in a retrieval pipeline that supplemented local model generation, reducing reliance on extensive parameter counts. Phi4, a 14-billion-parameter local LLM, was evaluated alongside general-purpose and medical-specialized LLMs, comparing its performance, energy efficiency, and monetary costs on healthcare-specific benchmarks. Phi4 achieved the highest accuracy of 33% while consuming 86.08 Joules per response, demonstrating its ability to rival GPT-4-1106-preview in accuracy with significantly lower energy and operational costs. Mid-sized local models, such as Gemma2-9B, provided a balance, achieving 68.24 Joules per response at slightly lower performance levels. In contrast, smaller local models, such as Llama3.2-3b, offered exceptional energy efficiency (2.26 Joules per response) but lacked the accuracy required for critical medical tasks.

The study also highlights the challenges of using medical-specialized LLMs, such as Meditron 7B and MedLLaMA 2, within a RAG framework. Meditron 7B failed to produce consistent responses, while MedLLaMA 2 struggled to adhere to prompts, rendering them ineffective for evaluation. These findings underscore the importance of alignment and optimization when deploying domain-specific models.

The dissertation concludes that integrating smaller, on-premises LLMs with a dedicated retrieval layer constitutes a viable strategy for delivering robust, contextually precise, and resource-efficient AI solutions in healthcare. Although fine-tuning protocols, retrieval parameter optimization, and energy efficiency warrant further investigation, the results demonstrate that locally deployed LLMs, such as Phi4, can effectively bridge the performance gap with commercial APIs, providing cost-effective, sustainable, and privacy-centric solutions adapted to the stringent requirements of the healthcare domain.

# Acknowledgements

This thesis represents the end of a long and challenging journey, and I could not have made it here without the support of so many wonderful people.

I am incredibly grateful to Professor Gil, João and Liliana for their invaluable guidance, for constantly challenging me, and for making my Fridays both enriching and filled with joy. Your encouragement and insights have been instrumental in shaping this work.

To my parents, who, even without always understanding my reasons, never clipped my wings, thank you. Your sacrifices, love, and unwavering belief in me have made this achievement possible.

To my sister, my lifelong companion and best friend, thank you for always being the person I can trust above all else. Your words always come at the right time and your presence has been my greatest comfort.

To my family, my greatest source of inspiration, thank you for the example of resilience and unity that you set every day. Your strength and love have shaped who I am.

To my friends from Taipas, Eduarda, Margarida, Eduardo, Nuno, and Chico, who have been by my side for so many years, thank you for your constant support and friendship.

To the incredible friends I made at university, who made this journey infinitely more beautiful: to my friends from JuniFEUP, AEFEUP, and Informática, thank you for giving so much meaning to my time at FEUP and making it feel like home.

To Margarida, my companion in every moment, my refuge and the keeper of my best memories from these years, I couldn't have done this without you. To Sílvia, my safe haven, for always giving your all for me and never letting me give up. I learn from you every day, and I am endlessly grateful for your constant presence, your strength, and your ability to find joy even in the face of challenges. And to Kika, Carol, and Inês, my greatest support system and the friends I will carry with me for life. I hope to continue sharing unforgettable moments with you, long beyond our university days. To Sofia, my first university friend and roommate, thank you for all the unforgettable moments and meaningful conversations, and to Mafalda, my most recent cousin, thank you for your support and for standing by my side, not only in the toughest moments but also in the most joyful ones.

To my boyfriend, Ricardo, who has been by my side through it all, thank you for your endless patience, for being my rock in the hardest moments, and for every sacrifice you made to help me get here. Your love and support have meant everything.

Finally, I dedicate this work to my avó Linda, who, unfortunately, never had the opportunity to pursue her studies, but always reminded me of how fortunate I was to have this chance. Her home was my first classroom, and I know that wherever she is, she's celebrating this achievement with a table full of cakes and rissóis, just as she always did.

Ana Luísa Ferreira Marques

*"O ser humano é só carne e osso*
*e uma tremenda vontade de complicar as coisas"*

Valter Hugo Mãe

# Contents

# List of Figures

# List of Tables

# Abreviaturas e Símbolos

| | |
|---|---|
| ACR | American College of Radiology |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| BLURB | Biomedical Language Understanding and Reasoning Benchmark |
| BO | Bayesian Optimization |
| CAD | Computer-Aided Design |
| CKD | Chronic Kidney Disease |
| CoT | Chain-of-Thought |
| DP | Differential Privacy |
| EHRs | Electronic Health Records |
| EM | Exact Match |
| FL | Federated Learning |
| HE | Homomorphic Encryption |
| HEQ | Human Equivalence Score |
| ICD | International Classification of Diseases |
| ICL | In-Context Learning |
| LLM | Large Language Model |
| LM | Language Model |
| LoRA | Low-Rank Adaptation |
| LOMO | Low-Memory Optimization |
| LSTM | Long Short-Term Memory |
| MAP | Mean Average Precision |
| MCQA | Multiple-Choice Question-Answering |
| MLM | Masked Language Modeling |
| MRR | Mean Reciprocal Rank |
| NDCG | Normalized Discounted Cumulative Gain |
| NER | Named Entity Recognition |
| NLP | Natural Language Processing |
| NLM | Neural Language Model |
| PEFT | Parameter-efficient fine-tuning methods |
| PLM | Prefix Language Modeling |
| PMC | PubMed Central |
| PTLM | Pre-trained Language Model |
| PUE | Power Usage Effectiveness |
| QA | Question Answering |

| | |
|---|---|
| RAG | Retrieval-Augmented Generation |
| RE | Relation Extraction |
| RLHF | Reinforcement Learning with Human Feedback |
| RNN | Recurrent Neural Network |
| SAS | Semantic Answer Similarity |
| SMC | Secure Multiparty Computation |
| SMS | Sentence Mover's Similarity |
| SoTA | State-of-The-Art |
| USMLE | United States Medical Licensing Examination |
| WMD | Word Mover's Distance |

# Chapter 1

# Introduction

This chapter presents an overview of the project by exploring its context, motivation, problem definition, and objectives. in which the project is inserted. Furthermore, it presents the overall document structure.

## 1.1 Context

Large Language Models (LLMs) have become central to Artificial Intelligence (AI) advancements due to their impressive abilities in understanding, generating, and reasoning. Incorporating LLMs into the healthcare field marks a pivotal advancement in AI applications to enable faster and more accurate diagnoses, continuous health monitoring, and the development of customized treatment plans.

LLMs such as GPT-4 and LLaMA exemplify advanced AI capabilities in understanding and generating human language and have significantly impacted sectors ranging from customer service to content creation. In healthcare, LLMs are used through various applications such as medical question-answering systems, patient data analysis, and support tools for healthcare providers. Oncology research is a notable example in which LLMs support tumour diagnosis tasks such as detection, sub-typing, staging, and grading. These applications emphasize LLMs' multimodal capabilities, as healthcare data often integrates text, images, and time series data. Using the strengths of LLMs, researchers and healthcare professionals can integrate multiple data modalities to improve diagnostic precision and patient care [1].

There is a growing interest in the integration of LLMs into healthcare, but despite the benefits of their applications, challenges such as interpretability, privacy, integration of medical knowledge, and collaboration with providers and patients complicate LLM implementation, so when incorporating LLMs into clinical practice, it must be considered [2].

## 1.2 Motivation

The use of LLMs, although promising, faces significant barriers due to the reliance on large service-based models, which involve high operational costs, substantial energy consumption, and elevated data

privacy risks. This issue is particularly critical in healthcare, where data sensitivity is crucial and strict regulatory standards demand robust data protection, creating the need for alternative approaches that preserve the advantages of AI while mitigating these issues. The primary motivation for this research is to address these limitations by employing LLMs that can offer comparable performance while overcoming these barriers.

Locally deployed LLMs represent a promising solution to handle these barriers effectively by reducing computational expenses and enabling data processing to remain on-premises, allowing the minimization of privacy risks and improving cost efficiency.

However, another critical limitation of LLMs is their inherent knowledge cap, as they are trained on static datasets up to a specific date. This constraint is particularly concerning in healthcare, where new research findings, treatment guidelines, and clinical protocols are continuously evolving. The dependency on outdated knowledge could alter decision making, underscoring the importance of integrating dynamic, up-to-date data sources into AI systems.

Emerging methods such as Retrieval-Augmented Generation present a compelling solution to these issues. RAG not only allows smaller models to achieve performance levels comparable to larger service-based systems but also provides a mechanism for dynamically retrieving the most current and relevant information. Demonstrating the effectiveness of this approach could lead to more scalable, privacy-conscious, and resource-efficient AI solutions in healthcare, ultimately making advanced AI tools accessible to a wider range of healthcare providers and improving patient care.

## 1.3 Problem Definition

This dissertation addresses a critical gap in the deployment of language models within healthcare applications. Existing service-based LLMs have shown proficiency in supporting healthcare tasks, but their dependence on large-scale infrastructure introduces high operational costs, substantial energy consumption, and increased privacy risks. Data sensitivity and compliance with HIPAA and GDPR regulations are essential in healthcare, making these limitations pose ethical and practical issues.

This dissertation focuses on the use of smaller-scale local LLMs enhanced with RAG for healthcare solutions. Many works have focused on the capabilities of large, service-based models without fully exploring the potential of smaller local models to maintain comparable performance levels. This research aims to identify potential improvements that could redefine the boundaries of effective and private AI in healthcare through the use of RAG techniques with local LLMs. It also focuses on minimizing both monetary and energy costs, striving to balance high model performance with ethical considerations related to sustainability and data privacy.

## 1.4 Objectives

The primary objective of this thesis is to evaluate the feasibility and effectiveness of deploying local, smaller-scale LLMs in healthcare applications, focusing on cost efficiency, energy consumption, and data privacy. Using RAG, these models may present a viable alternative to larger service-based LLMs, potentially maintaining high-performance standards in healthcare without using extensive resources or compromising data security. The research will focus on the following main objectives:

- Assess the performance of local LLMs in healthcare applications: Compare the performance of local, smaller-scale LLMs against large service-based LLMs in healthcare-specific tasks and discuss the privacy and security advantages of local LLMs
- Assess the performance of medical specialized LLMs in healthcare applications: Compare the performance of medical LLMs against general LLMs in healthcare-specific tasks
- Evaluate the cost and energy efficiency of local LLMs: Conduct a detailed analysis of the financial and energy demands associated with the deployment of local LLMs, comparing these factors with the requirements of larger service-based LLMs.

Given the problem described, we can define the hypothesis as follows:

- In healthcare applications, integrating RAG with local, smaller-scale language models can maintain performance levels comparable to large, service-based LLMs while significantly reducing monetary and energy costs and enhancing data privacy, establishing them as effective and efficient solutions.

Formally, these objectives can be condensed into the following set of research questions:

- **RQ1:** How can RAG be effectively integrated with smaller local LLMs to enhance their performance in healthcare applications, achieving results comparable to larger service-based models?
- **RQ2:** How do medical-specialized LLMs perform compared to general-purpose LLMs in addressing healthcare-specific tasks?
- **RQ3:** Do local LLMs offer significant reductions in monetary costs and energy consumption compared to large, service-based LLMs when used in healthcare settings?

The achievement of these objectives could provide valuable information on the viability of smaller local LLMs as cost-effective, secure, and energy-efficient alternatives to large service-based models for healthcare, potentially setting the stage for more sustainable and private AI solutions in healthcare.

## 1.5 Document Struture

This document is organized into seven more Chapters. Chapter 2 introduces the background knowledge, covering foundational concepts such as LLMs, RAG, privacy-preserving techniques, and the evolution of these technologies. Chapter 3 reviews the state-of-the-art, focusing on the applications of LLMs and RAG in healthcare, identifying the limitations of existing approaches, and highlighting the research gaps.

Chapter 4 outlines the implementation details of the proposed approach. Chapter 5 presents the testing setup and results, describing various scenarios tested. Chapter 6 provides a detailed analysis of the results, comparing the performance and trade-offs between local and service-based LLMs, while exploring the implications of these results. Finally, Chapter 7 concludes the thesis by summarizing the main contributions, and proposing potential directions for future work.

# Chapter 2

# Background

This chapter presents a technical background of the main fields required for a comprehensive understanding of the state-of-the-art and framework solution proposals in the following chapters. The chapter is divided into two sections. First, an explanation of Large Language Models, including their evolution, pre-training techniques and fine-tuning. Then, RAG paradigm is introduced along with the respective evaluation metrics.

## 2.1 Large Language Models

### 2.1.1 Terms and Definitions

Natural Language Processing (NLP) is a branch of artificial intelligence focused on the interaction between computers and human (natural) language. In this field, *language models* refer to computational models designed to process, understand, and generate human language.

Language Models (LM) are trained to estimate the likelihood of word sequences, which helps them perform tasks such as text generation, translation, and summarization by capturing contextual information across sentences or phrases to generate meaningful text. LMs can vary significantly in complexity and size, from simple statistical models to large neural-based systems with billions of parameters.

In neural networks, parameters are the internal variables (such as weights and biases) learned from the training data. They govern how input data is transformed into output predictions, determining the model's performance. Parameters are an important metric in assessing the size and capacity of language models: larger models, like GPT-3 with 175 billion parameters, can handle more complex linguistic structures.

Tokens are the basic units of text that language models process. Tokens can be words, subwords, characters, or symbols, depending on the model. For example, the word "understanding" could be tokenized into "under" and "standing" as subwords. Tokenization, the process of converting text into tokens, enables language models to process various languages and manage large vocabularies efficiently.

Embeddings are dense vector representations of tokens designed to capture semantic meaning by positioning words with similar meanings close to each other in a high-dimensional space. Embeddings allow language models to comprehend relationships between words, enhancing their ability to understand and generate contextually relevant text.

### 2.1.2 Evolution of Language Models

The development of language models has progressed through distinct phases each marked by significant innovations that improved their capability to understand and generate language as show in Figure 2.1

The earliest language models used statistical techniques, notably n-grams, sequences of words of length "n". For instance, a bigram model (n=2) would consider pairs of consecutive words to predict the next word based on its immediate context. Although n-gram models were foundational, they struggled with limitations such as data sparsity and an inability to capture long-term dependencies.

With the advent of neural networks, more sophisticated language models, such as Neural Language Models (NLMs), emerged. Recurrent Neural Networks (RNNs) [3] and Long Short-Term Memory (LSTM) [4] networks allowed language models to process sequences with awareness of the preceding context, thereby overcoming the limitations of the n-gram models. NLMs introduced embeddings, which allowed models to understand semantic relationships between words. However, RNNs and LSTMs faced challenges such as the vanishing gradient problem, which limited their effectiveness on longer text sequences.

The Transformer architecture was revolutionary for language modelling [5]. Unlike RNNs, transformers leverage a parallelized structure with a self-attention mechanism, allowing the model to assess the importance of different tokens across a sequence simultaneously rather than sequentially. This innovation improved computational efficiency and the model's ability to handle long-range dependencies, leading to the development of Pre-trained Language Models (PTLMs), such as BERT [6] and GPT [7], which could be trained on large corpora and fine-tuned on specific tasks for greater versatility.

LLMs are advanced LMs with a massive scale, often comprising billions of parameters and trained on extensive datasets. Building on transformers, the latest generation of LLMs, such as GPT-3 and PaLM, exhibit emergent abilities, including few-shot and zero-shot learning, allowing them to generalize tasks without extensive re-training. LLMs' increased size and capacity enable them to handle complex, multi-task applications, moving NLP toward creating versatile AI that can perform a broad range of language-related tasks.

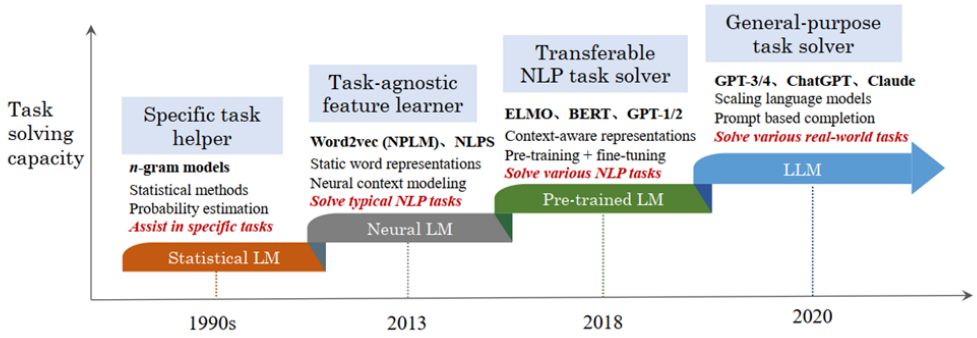


Figure 2.1: Evolution of LLMs

### 2.1.3 Transformer Architecture and Self-Attention Mechanism

The Transformer architecture is a foundational advancement in NLP, introducing a framework that allows language models to process entire sequences of tokens in parallel. At the core of the Transformer is the self-attention mechanism, which enables the model to weigh the importance of each token relative to others in a sequence. Each token is represented by three vectors: query, key, and value. These vectors are used to calculate an attention score, determining how much weight a token should give to other tokens in the sequence. This allows the model to capture the dependencies between distant tokens efficiently, enabling a deep understanding of the context.

To capture complex relationships between tokens, the Transformer model uses multi-head attention, where multiple attention heads operate in parallel, each with its own set of query, key, and value matrices. Each attention head focuses on different parts of the sequence, capturing diverse contextual relationships. The outputs of all attention heads are then concatenated and linearly transformed, giving the model a more nuanced understanding of token dependencies.

Unlike RNNs, transformers do not have an inherent mechanism to understand the order of tokens in a sequence, as they process tokens in parallel. To address this, positional encodings are added to each token's embedding, providing information about each token's position within the sequence. Positional encoding typically utilizes a sinusoidal function, enabling the model to differentiate tokens based on their position. This process captures sequential order and allows the model to grasp the syntactic structure.

Each Transformer layer includes a feedforward neural network applied to the output of the multi-head attention mechanism. Residual connections are also incorporated to prevent the vanishing gradient problem and ensure stable gradient flow during training. The use of residual connections, combined with layer normalization, helps maintain model stability, especially in deep architectures.

The mainstream architectures of existing LLMs can be categorized in encoder-only, decoder-only (casual and non-casual), and encoder-decoder models [8] [9].

Encoder-only LLMs, such as BERT and DeBERTa, use bidirectional training, allowing them to understand the context of a token by integrating information from both sides of the input sequence, making them ideal for tasks like sentiment analysis and document classification.

Decoder-only LLMs, including the GPT and LLaMA series, process text unidirectionally (left-to-right). They use a single text stream as input, and because of the causal masking pattern, conditioning is simply based on past tokens. Non-casual decoder models were proposed as a solution to allow decoder-only models to build richer representations of the input/conditioning text by modifying the attention mask used.

Encoder-decoder models, like Flan-T5 and ChatGLM, combine both approaches, using an encoder to understand input sequences and a decoder to generate output, making them suitable for tasks requiring both comprehension and generation of text.

LLMs are expanded versions of the Transformer architecture with increased layers, parameters, and larger pre-training datasets. The "Scaling laws" predict performance improvements as models grow in size, either by increasing parameters, layers, or training data. OpenAI's scaling laws suggest prioritizing model size over data volume for optimal performance, while Google DeepMind's scaling laws recommend increasing model and data sizes proportionally [10]. These principles guide researchers in resource allocation and in anticipating performance gains from scaling models.

### 2.1.4 Pre-training, Fine-tuning and Prompting Techniques

#### 2.1.4.1 Pre-training Approaches

A step in building LLMs is pre-training, where the model is trained on a large, unlabeled dataset via self-supervision [9]. Pre-training objectives play an important role in the model performance and include full language modelling, prefix language modelling, and masked language modelling, each with unique approaches to processing input and target tokens [11]

In Full Language modelling, the model learns to predict the next token based on a sequence of previous tokens, generating text one token at a time. This autoregressive training method has become standard for many large models, namely large decoder-only models, such as those in the GPT series, because it aligns closely with text generation tasks.

In Prefix Language Modeling (PLM) a "prefix" is set up within which the conditioning is non-causal, meaning the model isn't restricted to a strict left-to-right sequence for all tokens. The model predicts tokens outside the prefix, conditioned on all preceding tokens within the prefix. This objective allows the models to handle complex text dependencies and is used by encoder-decoder architectures and non-causal decoder-only models.

In Masked Language Modeling (MLM), certain tokens or spans of text within the input are replaced with a mask token, and the model is trained to predict these masked sections. BERT is a notable example of MLM [9].

#### 2.1.4.2 Fine-tuning Strategies

One limitation of pre-training LLMs on large, diverse text corpora is that these models may have difficulty capturing the specific distributional characteristics of particular task-related datasets. Fine-tuning

addresses this by further training the pre-trained model on a targeted dataset. This process can be achieved either by directly adjusting all model parameters with a language modelling objective or by adding learnable layers that make the model's representations compatible with specific tasks, such as text classification or sequence labelling [12].

Instruction tuning is a central approach, exemplified by models like GLM-130B and Galactica, which use instruction-formatted datasets during pre-training to improve task adaptability. Multi-stage instruction tuning further boosts conversational abilities while retaining task-specific knowledge by alternating between task and chat-focused tuning stages.

Alignment tuning through Reinforcement Learning with Human Feedback (RLHF), used in models like InstructGPT, aligns model outputs with human preferences by training on human feedback. This approach prioritizes safe, user-friendly responses and reduces the risk of generating harmful content [13].

Parameter-efficient fine-tuning methods (PEFT) include adapter and prefix tuning, which introduce small, task-specific modules to each Transformer layer without modifying the model's main parameters. These will be studied in detail in the next section.

### 2.1.5 Parameter Optimization Processes

Parameter optimization in LLMs is critical for improving their performance while managing computational resources efficiently. Several strategies and techniques have emerged to handle the challenges associated with optimizing the vast number of parameters in these models.

Full parameter fine-tuning involves adjusting all model parameters during the fine-tuning process. This is often resource-intensive, especially for large models, but allows for precise task-specific adaptations. Techniques like Low-Memory Optimization (LOMO) have been developed to make this process more efficient. LOMO fuses gradient computation and parameter updates to minimize memory overhead, enabling larger models to be fine-tuned with reduced hardware requirements, making the process feasible even with limited computational resources. This method eliminates the need to store all gradient tensors, significantly reducing memory usage. [14]

Hyperparameter optimization, which includes learning rates and batch sizes, is crucial for efficient model training. Bayesian Optimization (BO) is one of the common methods used for this purpose, leveraging surrogate models to predict promising hyperparameters for future trials. BO is particularly effective in scenarios where each hyperparameter evaluation is costly. However, it struggles with scalability when the number of hyperparameters grows. Alternative methods like gradient-based optimization are more scalable but often harder to implement. [15]

For parameter-efficient fine-tuning, Low-Rank Adaptation (LoRA) focuses on freezing most of the model's weights while introducing a low-rank decomposition of certain weight matrices. This allows tuning only a small fraction of the model's parameters, significantly reducing computational costs without sacrificing performance. LoRA has been shown to improve model performance in downstream tasks, making it particularly useful for adapting LLMs in resource-constrained environments. [16]

These methods collectively aim to balance performance with computational efficiency, ensuring that even large models can be optimized effectively without requiring excessive resources.

#### 2.1.5.1 Prompt Engineering

Prompt engineering is a critical method in optimizing LLMs to generate accurate, task-specific responses. This approach involves carefully designing input prompts to guide models in producing relevant and contextually appropriate outputs.

While fine-tuning is less computationally demanding than pre-training, it still requires additional model training and the availability of high-quality datasets, leading to the consumption of some computational resources and manual effort. In contrast, "prompting" methods efficiently adapt general LLMs (e.g., PaLM) to specialized domains, such as the medical field (e.g., MedPaLM), without the need for retraining model parameters. Common prompting techniques include In-Context Learning (ICL), Chain-of-Thought (CoT) prompting, Prompt Tuning, and RAG, allowing for effective alignment with minimal resource use.

ICL is designed to prompt a LLM to perform tasks efficiently by providing direct instructions. The ICL process involves four stages: task understanding, context learning, knowledge reasoning, and answer generation. First, the model interprets the specific goals and requirements of the task. Next, it grasps the relevant contextual information through the input provided. Then, the model utilizes its internal knowledge and reasoning abilities to recognize patterns and logic in the examples. Finally, it generates answers related to the task. Depending on the type and number of input examples, ICL is divided into three categories: One-shot Prompting, where only one example and task description are provided; Few-shot Prompting, which allows for multiple examples and task descriptions; and Zero-shot Prompting, which relies solely on task descriptions without examples. ICL enables LLMs to make task predictions based on context enriched with a few examples or demonstrations, allowing the model to learn from these examples and generate accurate responses.

CoT prompting improves model outputs' accuracy and logical coherence, surpassing ICL. [17] CoT encourages the model to generate intermediate steps or reasoning paths when solving complex tasks through specific prompts. This approach can also be combined with few-shot prompting by providing reasoning examples, allowing LLMs to include detailed reasoning processes in their responses. CoT has proven particularly effective for tasks requiring complex reasoning, such as medical question-answering, significantly improving model performance.

Prompt tuning seeks to improve the model performance by combining prompting and fine-tuning techniques [18]. This approach introduces learnable prompts: continuous, trainable vectors that can be optimized or adjusted during fine-tuning to better suit various medical tasks and scenarios. As a result, prompt tuning offers more flexibility in guiding LLMs compared to traditional methods that rely on static, fixed prompts. In contrast to conventional fine-tuning, which adjusts all model parameters, prompt tuning only modifies a small set of parameters directly related to the prompts, significantly reducing computational demands while maintaining accuracy.

RAG combines LLMs with external knowledge retrieval mechanisms. By integrating retrieved information directly into the prompt, RAG allows models to access more extensive datasets or updated knowledge bases. This approach will be further studied in the next section.

### 2.1.6 Service-based and Local LLMs

The core distinction between cloud-based LLMs and local LLMs lies in their deployment environments and their implications for data security, computational power, and customization.

Service-based LLMs are hosted on powerful remote servers provided by large tech companies like OpenAI, Google, or Amazon. Users access these models via APIs, leveraging cloud infrastructure capable of managing high computational workloads. In contrast, local LLMs are designed to run on-premises, either on personal devices or organizational servers, allowing for enhanced control and data privacy.

Service-based models benefit from near-limitless scalability due to centralized resources that can be scaled up to meet demand, thus supporting even the largest models like GPT-4 or PaLM. However, utilizing these models often requires sharing sensitive data over the internet, which raises privacy concerns. In contrast, local LLMs, such as LLaMA or Vicuna, keep all data processing local, mitigating risks associated with data exposure and making them more suitable for industries like healthcare, where privacy is crucial.

Local LLMs are highly customizable since users can modify model architecture and training parameters according to specific requirements. Fine-tuning or adapting local models with domain-specific data is generally easier compared to cloud models. Cloud-based LLMs, while accessible and easy to deploy, often have limitations in terms of direct model control, as users typically interact through predefined API interfaces with less flexibility to adapt the model itself.

In essence, cloud-based LLMs provide high performance and scalability, which is ideal for general applications requiring significant computing resources. Local LLMs, in contrast, offer enhanced privacy, control, and customizability, being well-suited for specialized tasks, especially where data confidentiality is critical and where custom fine-tuning can make a significant difference in task performance.

### 2.1.7 Privacy-Preserving Techniques

Privacy-preserving techniques are essential in handling sensitive data, especially within healthcare applications where personal information is protected under strict regulations. These methods ensure that data used by models remains confidential, even during analysis, training, and deployment stages. Prominent techniques include Federated Learning, Differential Privacy, and Homomorphic Encryption combined with Secure Multiparty Computation (SMC). Federated learning allows models to be trained across decentralized devices, such as hospital systems, without direct data sharing. Each device trains the model locally, contributing only encrypted or aggregated updates, thus preventing raw data from leaving local

environments. This technique enables model development using diverse data sources while maintaining data confidentiality and is well-suited to multi-institutional healthcare systems.

Differential privacy adds noise to data or model parameters, protecting individual privacy even in statistical analysis or model updates. This approach has gained traction in sensitive domains, such as healthcare, by making it difficult to identify individuals from model outputs. Furthermore, homomorphic encryption allows computations on encrypted data, supporting operations without decrypting sensitive information. When combined with Secure Multiparty Computation, homomorphic encryption enables collaborative model training across parties, maintaining privacy across all stages of the process, which is especially critical in healthcare collaborations where data sharing is tightly regulated.

Federated Learning (FL) offers a decentralized approach to training machine learning models, leveraging data from multiple locations without centralizing it. In healthcare, this approach is beneficial as it allows hospitals, research institutions, and clinics to collaboratively train models on diverse datasets without violating patient privacy regulations. Each institution processes its data locally, contributing encrypted model updates to a central model without transferring raw data. Applications of federated learning in healthcare are vast, spanning from predictive modelling of disease outbreaks to personalized medicine and medical imaging analysis. The benefit of FL is that it enables richer, more generalized models trained on diverse, real-world data while protecting patient privacy.

Differential Privacy (DP) is a framework that ensures privacy by adding statistical noise to data or model outputs, which obscures individual-level information in aggregated datasets. This method allows insights to be derived from data while maintaining robust privacy guarantees, as it prevents reverse-engineering of personal data from the model outputs. DP techniques are effective in healthcare, where patient data sensitivity is paramount. Applications of differential privacy include creating public datasets from private data and training models without exposing sensitive information. Effectiveness in healthcare research has been demonstrated through improvements in machine learning privacy-preserving capabilities, making DP a valuable tool for both research and deployment of models in clinical settings.

Homomorphic Encryption (HE) allows computations on encrypted data without decryption, providing a robust privacy-preserving framework for collaborative machine learning. HE supports operations such as addition and multiplication on ciphertexts, enabling complex computations directly on encrypted data. This approach is particularly advantageous for multi-institutional collaborations, where sensitive data can be used in model training without direct sharing. When combined with SMC, HE enables encrypted data from multiple sources to be processed jointly, with each party performing computations on their encrypted data. This setup ensures that data privacy is maintained throughout the training and inference processes, making it ideal for healthcare applications, where stringent privacy requirements demand secure, collaborative methods.

## 2.2 Retrieval-Augmented Generation

### 2.2.1 Concept and Mechanism of RAG

RAG represents a paradigm that combines a retrieval component with a generative model to improve response quality, particularly for tasks requiring access to extensive, domain-specific knowledge. RAG allows LLMs to retrieve relevant information from external knowledge sources, such as databases, to generate more accurate and contextually relevant outputs.

The process involves three core steps: indexing, retrieval, and generation. In indexing, raw documents are transformed into vector representations that capture the semantic meaning of each piece of information. Techniques like Sentence-BERT are commonly used to create and organize these embeddings that are stored in a vector database. This allows the retrieval system to search through vast datasets rapidly and locate information that aligns semantically with the user's query. Indexing ensures that the retrieval system can efficiently handle large-scale data and support real-time interactions.

During retrieval, the RAG system identifies and selects the most relevant information for a given input query. When a query is entered, it is transformed into a vector representation using the same embedding technique as the indexed data. The retrieval system then searches the vector database for entries with high semantic similarity to the query, using similarity metrics like cosine similarity or Euclidean distance to rank potential matches. The top-k most relevant chunks (e.g., paragraphs or passages) are then passed to the generative model. This retrieval step is crucial for grounding the model's response in current or domain-specific knowledge, particularly valuable in cases where the LLM's pre-trained data may be outdated or incomplete.

In the final stage, the retrieved data is combined with the original query as input for the generative model. The model, typically a transformer-based LLM, integrates the retrieved context to produce a coherent, contextually accurate response. The generation process involves attention mechanisms that allow the model to weigh and incorporate relevant parts of the retrieved text while responding. This step generates a response that not only aligns with the query but is also enriched with precise, up-to-date, and domain-relevant information from the external source.

To refine the retrieval and generation processes, some RAG systems use advanced retrieval techniques like multi-step retrieval and ranking. In multi-step retrieval, the model iteratively refines its search based on intermediate results, improving accuracy by narrowing down the information with each step. Ranking, on the other hand, involves scoring and filtering retrieved items based on relevance to ensure only the most pertinent information is included in the final response. Techniques like cross-attention between the query and retrieved documents during the generation stage also improve the integration of retrieved content, helping the model maintain coherence and relevance in its response.

In [19] the authors systematically review RAG's technical evolution, categorizing it into three main paradigms: Naive RAG, which relies on basic retrieval and generation processes; Advanced RAG, which incorporates refined methods like reranking and filtering to improve retrieval relevance; and Modular

RAG, where distinct components (retrieval and generation) are more deeply integrated and potentially co-optimized for better performance across tasks.

### 2.2.2 Benefits and Applications of RAG

RAG offers distinct benefits by integrating real-time external data retrieval with generative language models. This combination enhances the quality, accuracy, and relevance of model outputs, especially in knowledge-intensive tasks where static model parameters may otherwise limit performance.

One of RAG's primary benefits is its improved factual accuracy. By drawing on up-to-date or domain-specific external sources, RAG can address limitations associated with LLMs, such as hallucinations—where models generate inaccurate information based on gaps in their pre-trained knowledge. For example, RAG has effectively produced factually correct responses for open-domain question-answering systems.

RAG's architecture enhances relevance and specificity in model responses, which is particularly useful for handling rare or unseen data. For instance, models augmented with RAG are better at accessing domain-specific knowledge on demand, a significant advantage in areas like healthcare and law, where up-to-date or specialized information is crucial. This mechanism helps overcome the limitations of pre-trained LLMs, which may lack comprehensive domain-specific details due to the general nature of their training corpora.

In practical applications, RAG enables dynamic adaptability across fields that require frequently updated data, such as customer support, scientific research, and content generation. In customer service chatbots, RAG can access databases of user queries, allowing models to generate contextually accurate responses grounded in the latest available data. In research, RAG supports real-time retrieval of scientific literature, enabling applications that assist with summarization, literature review, and academic inquiry to provide high-relevance answers.

RAG has also been shown to contribute significantly to reducing hallucination rates in complex conversational and decision-making tasks, providing reliable information sourced directly from external repositories. This characteristic makes RAG particularly beneficial in applications where factual reliability is non-negotiable, such as medical diagnosis support or legal consultations.

Additionally, by reducing the dependency on continuous model fine-tuning, RAG optimizes computational resources, making it a cost-effective alternative for systems requiring frequent knowledge updates.

### 2.2.3 Parameter Optimization

Parameter optimization in RAG is a crucial step to ensure that both retrieval and generation components work synergistically to provide accurate and contextually relevant responses. In a RAG framework, a retriever model identifies relevant documents from a large corpus based on a query, while a generator model, often a large language model, synthesizes this information to produce a coherent answer. Key

parameters to optimize include the retriever's top-k selection threshold, which dictates how many relevant documents are retrieved, and the similarity threshold, which influences the relevance of selected documents. Tuning these parameters balances retrieval precision with recall, affecting the generation model's input quality. Similarly, optimizing generation parameters like maximum token length, temperature, and beam search width in the generation model impacts response specificity and fluency, ensuring generated answers remain concise yet informative.

Additionally, hyperparameters related to embedding dimension and retrieval model architecture can significantly impact the retriever's ability to capture semantic relevance in retrieved documents. Advanced optimization methods, such as grid search, Bayesian optimization, or reinforcement learning, are commonly used to identify the ideal parameter configurations. In a RAG system, parameter optimization must consider real-world constraints, such as latency and computational costs, especially if the system is deployed in real-time applications. Regular evaluation using metrics like BLEU, ROUGE, and F1 for language generation, as well as recall and mean average precision (MAP) for retrieval, helps refine these parameters iteratively. As a result, well-optimized RAG systems demonstrate enhanced answer accuracy, faster response times, and overall improvement in user experience, making parameter optimization a foundational process for high-performing retrieval-augmented models.

### 2.2.4 Evaluation of RAG

For the retrieval stage, evaluation metrics such as Hit Rate, Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) are commonly employed. These metrics assess the retrieval system's effectiveness in sourcing relevant information, which is critical for generating accurate responses. In the generation stage, traditional metrics such as BLEU and ROUGE scores are used to evaluate the coherence of the content. In contrast, accuracy and Exact Match (EM) scores are applied to determine the faithfulness of responses. Precision and recall measure how well the generated outputs align with the retrieved context.

The RAG evaluation framework outlined in [19] highlights three quality aspects: context relevance, answer faithfulness, and answer relevance. Context relevance measures how closely the retrieved data matches the query, ensuring minimal noise in the input for generation. Answer faithfulness evaluates whether the generated response accurately reflects the retrieved content, helping to avoid contradictions. Answer relevance assesses the direct alignment of the generated response with the query, verifying that the answer addresses the posed question directly.

Benchmarks such as RGB, RECALL, and CRUD are commonly used to provide standardized testing for RAG models, allowing researchers to gauge model performance on various retrieval and generation tasks. Automated tools, including RAGAS, ARES, and TruLens, support these evaluations by analyzing and scoring RAG outputs systematically, thus enabling a scalable approach to quality measurement.

# Chapter 3

# State-of-the-Art

Through a comprehensive exploration of both Large Language Models, Retrieval-Augmented Generation and Performance, Energy, and Cost Metrics in LLMs in healthcare, this chapter provides an overview of the current state-of-the-art methodologies and techniques.

## 3.1 Large Language Models in Healthcare

Integrating LLMs into healthcare has opened new horizons for medical research, diagnostics, and patient care. LLMs, trained on vast amounts of text data, can understand and generate human-like language, making them valuable for various applications in MedTech.

This section explores the different applications of LLMs in healthcare and reviews the models used in various studies, both general-purpose and medical-specialized LLMs.

### 3.1.1 General Large Language models

LLMs have made significant strides due to their capacity for understanding, generating, and reasoning over complex texts. These models rely on large-scale datasets during their pre-training phases. For instance, the CommonCrawl, Wikipedia, and Books datasets are often used, containing diverse types of textual data that enable the models to generalize across many domains. In Table 3.1 you can see an overall comparison of general LLMs.

| **Model** | **Structure** | **Number of Parameters** | **Pre-training Data Scale** | **Use Case** |
|---|---|---|---|---|
| GPT-3 [11] | Decoder-only | 6.7B / 13B / 175B | 300B tokens | General-purpose NLP, customer service, text generation |
| GPT-4 [7] | Decoder-only | - | - | Advanced conversational AI, multi-modal tasks |
| PaLM [20] | Decoder-only | 8B / 62B / 540B | 780B tokens | Medical research, multilingual NLP |
| FLAN-PaLM [21] | Decoder-only | 540B | - | Question answering, instruction-following tasks |
| Claude-3 [22] | Decoder-only | - | - | Customer support, general text generation |
| Gemini (Bard) [23] | Decoder-only | - | - | Interactive Q&A, dialogue systems |
| UL2 [24] | Encoder-Decoder | 19.5B | 1T tokens | Text understanding and generation |
| T5 [25] | Encoder-Decoder | 11B | 1T tokens | Language translation, summarization, text classification |
| Vicuna [26] | Decoder-only | 7B / 13B | LLaMA + 70K dialogues | Chatbots, fine-tuned dialogue systems |
| Alpaca [27] | Decoder-only | 7B / 13B | LLaMA + 52K IFT | Conversational agents, personal assistants |
| LLaMA [28] | Decoder-only | 7B / 13B / 33B / 65B | 1.4T tokens | On-device language understanding, specialized NLP tasks |
| LLaMA-2 [29] | Decoder-only | 7B / 13B / 34B / 70B | 2T tokens | Language modeling with domain-specific customization |
| LLaMA-3 [30] | Decoder-only | 8B / 70B | 15T tokens | Local NLP applications with enhanced context understanding |
| Mistral [31] | Decoder-only | 7B | - | On-device generation for low-resource tasks |
| ChatGLM [32] | Encoder-Decoder | 6.2B | 1T tokens | Local multilingual processing, interactive AI |
| BERT [6] | Encoder-only | 110M / 340M | 3.3B tokens | Contextual understanding, document analysis |
| DeBERTa [8] | Encoder-only | 1.5B | 160GB | Text classification, language understanding |
| RoBERTa [33] | Encoder-only | 355M | 161GB | Sentence prediction, comprehension tasks |

Table 3.1: Comparison of General LLMs

In healthcare, general-purpose LLMs, such as GPT-4 and ChatGPT, have been adapted to tackle various tasks, including diagnostics, patient interaction, and medical literature summarization. These models process clinical data such as patient records, generate diagnostic reports, and respond to medical inquiries.

Nori *et al.* [34] evaluated the performance of GPT-4 on medical tasks, particularly through its performance on the U.S. Medical Licensing Examination (USMLE) and the MultiMedQA benchmark suite. The study reveals that GPT-4, without specialized medical training, achieves high scores across several benchmark datasets, surpassing task-specific models like Flan-PaLM 540B and Med-PaLM in all areas except PubMedQA. The results demonstrate GPT-4's robustness in multiple-choice medical tasks with zero-shot and five-shot settings. This model also shows improved calibration over GPT-3.5, meaning it better aligns its confidence levels with actual answer correctness, crucial in high-stakes medical applications where reliable confidence estimation is essential. While GPT-4 performs well on structured exams, its real-world deployment in clinical settings requires careful assessment to mitigate risks related to accuracy and ethical considerations.

Ankit Pal and Malaikannan Sankarasubbu [35] evaluated Google's multimodal LLM, Gemini, within the medical field. Gemini was tested on multiple medical benchmarks, including MultiMedQA (for reasoning), Med-HALT (for hallucination detection), and Medical VQA. While the model demonstrated solid medical knowledge, it fell short in diagnostic accuracy when compared to models like Med-PaLM 2 and GPT-4, particularly in interpreting complex visual medical data. It performed lower than GPT-4 in VQA tasks, achieving an accuracy of 61.45% compared to GPT-4V's 88%. The results of the evaluation highlighted Gemini's susceptibility to hallucinations (i.e., generating incorrect or unfounded information), which poses challenges in high-stakes medical settings.

In [36], researchers investigate the performance and limitations of Mistral's LLM, Mistral-Large-2402, in medical applications. They examined Mistral's ability to handle complex medical reasoning tasks, including diagnosis, treatment recommendations, and condition analysis, while also assessing its tendency to produce hallucinations. Mistral-Large-2402 was evaluated using the MedQA, PubMedQA and MedMCQA datasets, with accuracy, coherence, and response completeness as key metrics for assessing the model's reliability in generating medical advice. In addition to its raw performance, the paper examined how the model handles factual consistency, especially its capacity to avoid making unsubstantiated claims or to indicate limitations when confronted with questions beyond its expertise. Techniques like few-shot prompting, self-consistency and RAG were used to enhance Mistral's performance and reduce the likelihood of hallucinations. The results revealed Mistral-Large-2402's strengths in addressing medical questions, with 77,5% of accuracy in the PubMedQA dataset. The model showed an average accuracy increase of 6.45% over MedAlpaca 7B, 18.05% over MediTron-7B, and a notable 31.12% improvement over PMC-LLaMA 7Bning.

In [30] evaluated the performance of Llama 3 against leading proprietary models. The study assessed model performance using two sets of questions: the 2022 American College of Radiology (ACR) in-training exam and 85 additional radiology board-style questions unseen by the models. The results

showed that Llama 3 70B performed competitively with proprietary models while answering high-stakes medical questions by achieving 74% accuracy on the ACR in-training exam and 80% on the radiology board-style questions. These scores were close to the performance of GPT-4 Turbo, which achieved, respectively, 78% and 82% on the same questions. Additionally, this model outperformed smaller open-source models (such as Mixtral $8 \times 7$B and Llama 3 8B). This research demonstrates the growing potential of open-source models in healthcare and radiology, providing a privacy-respecting alternative with competitive performance.

Despite their potential, general LLMs sometimes lack the domain-specific knowledge required for accurate medical interpretations. They may generate plausible but incorrect medical information, posing risks in clinical applications [11].

### 3.1.2 Specialized Medical LLMs

Specialized medical LLMs have emerged to address the limitations of general-purpose LLMs. These LLMs are primarily developed by either pre-training from scratch, fine-tuning existing general LLMs, or using prompting techniques to adapt general LLMs to the medical domain. In Table 3.2 we provide a comparison overview of Medical-domain LLMs.

Pre-training typically involves training an LLM on a large corpus of medical texts, including both structured and unstructured text, to learn the rich medical knowledge [2]. The corpus may include EHRs, clinical notes, and medical literature. PubMed, MIMIC-III clinical notes, and PubMed Central (PMC) are three of the most used medical corpora for LLM pre-training. Some models are trained on a single dataset, such as ClinicalBERT [37] (pre-trained on MIMIC-III), while others, like BioBERT [38], combine multiple sources, being pre-trained on both PubMed and PMC.

Training a medical LLM from scratch is costly and time-intensive, often requiring significant computational resources over extended periods [39]. An efficient alternative is to fine-tune general LLMs with medical data, enabling them to acquire domain-specific knowledge without starting from the ground up. Examples include MedPaLM, which was designed to handle complex clinical dialogues and achieve high accuracy on medical question-answering benchmarks such as the USMLE. This fine-tuning process enhances the models' relevance to healthcare applications, improving their ability to handle complex medical questions and generate detailed clinical insights.

Fine-tuning is less computationally intensive than pre-training but still requires additional training and high-quality datasets, which consume resources and demand manual labor. In contrast, prompting methods offer a more resource-efficient approach by adapting general LLMs to medical contexts without modifying model parameters. For instance, MedPaLM-2 [39] employs CoT prompting to solve medical reasoning tasks, which enhances its ability to provide structured and reliable answers in medical contexts.

Table 3.2: Comparison of Medical-Domain LLMs

| **Model** | **Development** | **Number of Parameters** | **Data Scale** | **Data Source** |
|---|---|---|---|---|
| BioBERT [38] | Pre-training | 110M | 18B tokens | PubMed, PMC |
| PubMedBERT [40] | Pre-training | 110M / 340M | 3.2B tokens | PubMed, PMC |
| SciBERT [41] | Pre-training | 110M | 3.17B tokens | Literature |
| NYUTron [42] | Pre-training | 110M | 7.25M notes, 4.1B tokens | NYU Notes |
| ClinicalBERT [37] | Pre-training | 110M | 112k clinical notes | MIMIC-III |
| BioM-ELECTRA [43] | Pre-training | 110M / 335M | - | PubMed |
| BioMed-RoBERTa [44] | Pre-training | 125M | 7.55B tokens | S2ORC |
| BioLinkBERT [45] | Pre-training | 110M / 340M | 21GB | PubMed |
| BlueBERT [46] | Pre-training | 110M / 340M | >4.5B tokens | PubMed, MIMIC-III |
| SciFive [47] | Pre-training | 220M / 770M | - | PubMed, PMC |
| ClinicalT5 [48] | Pre-training | 220M / 770M | 2M clinical notes | MIMIC-III |
| MedCPT [49] | Pre-training | 330M | 255M articles | PubMed |
| DRAGON [50] | Pre-training | 360M | 6GB | BookCorpus |
| BioGPT [51] | Pre-training | 1.5B | 15M articles | PubMed |
| BioMedLM [52] | Pre-training | 2.7B | 110GB | Pile |
| OphGLM [53] | Pre-training | 6.2B | 20k dialogues | MedDialog |
| GatorTron [54] | Pre-training | 8.9B | >82B tokens | EHRs, PubMed, Wiki, MIMIC-III |
| GatorTronGPT [55] | Pre-training | 5B / 20B | 277B tokens | EHRs |
| DoctorGLM [56] | Fine-tuning | 6.2B | 323MB dialogues | CMD |
| BianQue [57] | Fine-tuning | 6.2B | 2.4M dialogues | BianQueCorpus |
| ClinicalGPT [58] | Fine-tuning | 7B | 96k EHRs + 100k dialogues | MD-EHR, MedDialog, VariousMedQA |
| Qilin-Med [59] | Fine-tuning | 7B | 3GB | ChiMed |
| ChatDoctor [60] | Fine-tuning | 7B | 110k dialogues | HealthCareMagic, iCliniq |
| BenTsao [61] | Fine-tuning | 7B | 8k instructions | CMeKG-8K |

| **Model** | **Development** | **Number of Parameters** | **Data Scale** | **Data Source** |
|---|---|---|---|---|
| HuatuoGPT [62] | Fine-tuning | 7B | 226k instructions | Hybrid SFT |
| Baize-healthcare [63] | Fine-tuning | 7B | 101K dialogues | Quora, MedQuAD |
| BioMedGPT [64] | Fine-tuning | 10B | >26B tokens | S2ORC |
| MedAlpaca [65] | Fine-tuning | 7B / 13B | 160k medical QA | Medical Meadow |
| AlpaCare [66] | Fine-tuning | 7B / 13B | 52k instructions | MedInstruct-52k |
| Zhongjing [67] | Fine-tuning | 13B | 70k dialogues | CMtMedQA |
| PMC-LLaMA [68] | Fine-tuning | 13B | 79.2B tokens | Books, Literature, MedC-I |
| CPLLM [69] | Fine-tuning | 13B | 109k EHRs | eICU-CRD, MIMIC-IV |
| Med42 | Fine-tuning | 7B / 70B | 250M tokens | PubMed, MedQA, OpenOrca |
| MEDITRON [70] | Fine-tuning | 7B / 70B | 48.1B tokens | PubMed, Guidelines |
| OpenBioLLM [71] | Fine-tuning | 8B / 70B | - | - |
| MedLlama3-v20 [72] | Fine-tuning | 8B / 70B | - | - |
| Clinical Camel [73] | Fine-tuning | 13B / 70B | 70k dialogues + 100k articles | ShareGPT, PubMed, MedQA |
| MedPaLM-2 [74] | Fine-tuning | 340B | 193k medical QA | MultiMedQA |
| Med-Flamingo [75] | Fine-tuning | - | 600k pairs | Multimodal Textbook, PMC-OA, VQA-RAD, PathVQA |
| LLaVA-Med [76] | Fine-tuning | - | 660k pairs | PMC-15M, VQA-RAD, SLAKE, PathVQA |
| MAIRA-1 [77] | Fine-tuning | - | 337k pairs | MIMIC-CXR |
| RadFM [78] | Fine-tuning | - | 32M pairs | MedMD |
| Med-Gemini [79] | Fine-tuning | - | - | MedQA-R&RS, MultiMedQA, MIMIC-III, MultiMedBench |
| CodeX [80] | Prompting | GPT-3.5 / LLaMA-2 | CoT | - |
| DeID-GPT [81] | Prompting | ChatGPT / GPT-4 | CoT | - |
| ChatCAD [82] | Prompting | ChatGPT | ICL | - |
| Dr. Knows [83] | Prompting | ChatGPT | ICL | UMLS |
| MedPaLM [39] | Prompting | PaLM (540B) | CoT & ICL | MultiMedQA |

| **Model** | **Development** | **Number of Parameters** | **Data Scale** | **Data Source** |
|---|---|---|---|---|
| MedPrompt [84] | Prompting | GPT-4 | CoT & ICL | - |
| Chat-Orthopedist [85] | Prompting | ChatGPT | RAG | PubMed, Guidelines, UpToDate, Dyname |
| QA-RAG [86] | Prompting | ChatGPT | RAG | FDA QA |
| Almanac [87] | Prompting | ChatGPT | RAG & CoT | Clinical QA |
| Oncology-GPT-4 [88] | Prompting | GPT-4 | RAG & ICL | Oncology Guidelines from ASCO and ESMO |

The integration of LLMs into the medical field presents considerable promise in several domains. For diagnostics, LLMs like MedPaLM-2 achieve high accuracy in clinical assessments and question-answering, significantly improving patient outcomes by offering precise diagnostic suggestions.

[79] introduced Med-Gemini, a family of advanced multimodal models built upon Gemini, fine-tuned for medical applications to address complex clinical reasoning, multimodal understanding, and long-context processing. Med-Gemini achieves state-of-the-art (SoTA) performance on 10 out of 14 benchmarks, surpassing GPT-4 and Med-PaLM 2 on tasks such as USMLE (MedQA) with a 91.1% accuracy. Med-Gemini also has great results in multimodal tasks, such as interpreting medical images and supporting radiology workflows, showing potential for real-world utility in medical summarization, referral letter generation, and diagnostic dialogues.

[39] presented Med-PaLM 2, an advanced large language model specifically designed for medical question answering. Med-PaLM 2 builds on Google's PaLM 2 model and uses extensive medical domain fine-tuning and novel prompting strategies, including an ensemble refinement approach, to enhance medical reasoning capabilities. This model demonstrates state-of-the-art performance on multiple medical benchmarks, including the MedQA (USMLE-style questions), MedMCQA, and PubMedQA datasets. Med-PaLM 2 achieves a high accuracy rate of 86.5% on the MedQA dataset, a substantial improvement over its predecessor, Med-PaLM. The model also shows competitive or superior performance compared to existing models on other datasets. The paper also highlights Med-PaLM 2's improved performance in long-form medical question answering, with evaluations indicating that physicians preferred Med-PaLM 2's answers over those provided by physicians themselves in terms of clinical utility on several key axes, such as medical consensus alignment and reasoning quality.

[51] presents BioGPT, a generative pre-trained language model developed specifically for the biomedical domain by Microsoft Research. Unlike previous BERT-based models, which mainly focused on understanding tasks, BioGPT uses the GPT architecture, making it suitable for text generation. The model was pre-trained on large biomedical datasets, primarily PubMed abstracts, and evaluated across multiple tasks, including relation extraction, question-answering, document classification, and text generation. BioGPT established a high standard for biomedical QA tasks by outperforming state-of-the-art BioLinkBERT by 6.0%, with an accuracy of 78.2%.

In [89], researchers evaluated various models, including MedLLaMA-13B performance across several medical NLP tasks, such as named entity recognition, relation extraction, and question answering, using the Biomedical Language Understanding and Reasoning Benchmark (BLURB). MedLLaMA was effective in question-answering tasks, achieving competitive results compared to general-purpose models, especially when leveraging example-based prompting to enhance accuracy. However, it performed less consistently on other tasks, such as document classification, indicating the need for prompt customization to maximize its adaptability for various medical applications.

[90] introduced MEDITRON-70B, a suite of medical language models (7B and 70B parameters) adapted from LLaMA-2 for healthcare applications. The authors designed MEDITRON to democratise medical knowledge access using a pre-training dataset of clinical guidelines, PubMed abstracts, and experience replay data. The models were evaluated using significant medical reasoning benchmarks, including MedQA, MedMCQA, PubMedQA, and MMLU-Medical. MEDITRON-70B achieved a 70.2% accuracy on MedQA's 4-option USMLE-style questions (MedQA-style), which brought it within 5% of GPT-4's performance and 10% of Med-PaLM 2, two much larger commercial models. Furthermore, this model outperformed large open-source models like LLaMA-2 and PMC-LLaMA on medical benchmarks by up to 12% after incorporating advanced prompting methods such as chain-of-thought and self-consistency. The model was trained using a distributed framework on 128 Nvidia A100 GPUs, employing, among others, tensor and pipeline parallelism to efficiently process the massive medical corpus, highlighting the model's scalability.

[38] discussed BioBERT, a biomedical language representation model developed by adapting BERT specifically for biomedical text mining tasks. BioBERT was pre-trained on vast biomedical corpora, including PubMed abstracts and PMC full-text articles. The pre-training enhances BERT's understanding of biomedical terminology and relationships, making BioBERT better suited for tasks like named entity recognition (NER), relation extraction (RE), and biomedical question answering (QA). The results showed that BioBERT outperforms BERT and previous state-of-the-art models across these tasks. For example, in question answering, the evaluation was conducted using Biomedical Semantic QA datasets (which provide factoid and list questions based on PubMed); BioBERT achieved an MRR of 45,5%, 12.24% improvement compared to BERT.

[91] presents BioMistral, an open-source LLM specifically tailored for the biomedical domain. Developed by further training the Mistral 7B Instruct model on medical texts from PubMed Central, BioMistral aims to provide a versatile, multilingual, and efficient tool for medical applications, addressing the limitations of proprietary models regarding accessibility and data privacy. The results demonstrated that this BioMistral outperforms the base Mistral 7B and other open-source biomedical models across various medical question-answering tasks, particularly excelling in few-shot and fine-tuned scenarios. Supervised fine-tuning allowed BioMistral to surpass baseline performance on most benchmarks, including MedQA and PubMedQA. Additionally, model merging techniques, especially SLERP, improved accuracy by over 5%, while quantization reduced computational demands with minimal impact on accuracy. Although BioMistral showed some limitations in multilingual tasks, particularly against

GPT-3.5 Turbo, it proved to be a strong, accessible alternative for medical NLP tasks in English.

### 3.1.3 Applications of LLMs in Healthcare

#### 3.1.3.1 Medical tasks

There are two main types of medical machine learning tasks: discriminative and generative. Discriminative tasks involve categorizing or extracting information, such as question answering, entity extraction, relation extraction, text classification, and information retrieval. These tasks enable systems to organize and interpret medical data, aiding clinicians in diagnosis and decision-making. For instance, entity extraction helps identify critical medical terms, while question-answering provides precise responses based on clinical knowledge [39].

Generative tasks, on the other hand, require models to produce a coherent new text, such as medical text summarization, generation, and simplification. Summarization condenses long medical documents, aiding in quick comprehension. Text generation provides discharge instructions or health advice, and simplification translates complex medical jargon for patient understanding [92]

[2] performs a comparison demonstrating that general LLMs can perform great on existing medical machine-learning tasks. Shows that GPT-4 excels in closed-ended QA tasks, often surpassing task-specific models and approaching human performance. However, it underperforms in the open-ended and non-QA tasks, where task-specific models, such as BioBERT, achieve higher accuracy, indicating that LLMs still require improvements to support clinical applications fully. Expanding evaluations to broader tasks beyond QA is necessary to ensure LLMs are viable for real-world clinical decision-making.

#### 3.1.3.2 Clinical applications

AI technologies are transforming healthcare by supporting clinicians in making informed decisions, improving patient interactions, generating clinical reports, and providing mental health support [2].

One notable application is in medical decision-making, where LLMs have been deployed to analyze extensive datasets, including medical literature and patient histories, to suggest accurate diagnoses and personalized treatment options. For instance, models like NYUTron [42], trained on vast amounts of hospital data from New York University, support clinicians by predicting patient outcomes, including in-patient mortality prediction, comorbidity index prediction and readmission prediction, and streamlining operational tasks such as anticipating insurance claim denials. Another example, TrialGPT [93], leverages the predictive capabilities of GPT models to assist in clinical trial matching by providing criterion-specific eligibility assessments, reducing human screening time while enhancing accuracy. This model supplies clinicians with relevant trial information and explanatory context, facilitating faster and more precise decision-making.

One limitation of using LLMs for medical diagnosis is their dependence on subjective text inputs from patients, as they cannot directly interpret diagnostic images essential for objective diagnoses. Instead, LLMs can complement vision-based models by providing logical reasoning to enhance diagnostic

accuracy. An example is ChatCAD [82], where images are processed by a computer-aided diagnosis (CAD) model, and the resulting data is converted into text for ChatCAD to interpret and generate diagnoses.

LLMs can have a crucial role in facilitating therapeutic interactions and providing patients with preliminary support in cases of mental health support. PsyChat and Psy-LLM are two great examples of this use case. The first one, PsyChat, [94] is a client-focused LLM dialogue system that offers psychological support through five core modules: client behaviour recognition, counsellor strategy selection, input packaging, response generation (fine-tuned using ChatGLM-6B with an extensive dialogue dataset), and response selection. This system's effectiveness and practicality in real-world mental health support scenarios have been validated by both automated and human evaluations. Psy-LLM [95] is a model designed as a tool to support the workflow of professional counsellors, particularly in aiding individuals who may be experiencing depression or anxiety. These AI-driven dialogues bridge the accessibility gaps in mental health services by offering patients guidance and suggesting coping strategies.

Two of the biggest challenges in using LLMs for mental health support are their limited emotional understanding and the risk of producing harmful responses. LLMs often struggle to interpret complex emotional states and cannot provide the empathy essential in therapy. Additionally, without proper training and controls, LLMs may generate responses that are insensitive, inappropriate, or not based on evidence-based mental health practices, potentially harming individuals in vulnerable states. Addressing these issues requires rigorous, ethical training of LLMs and collaboration between AI researchers and mental health professionals. [96]

AI-powered assistants, such as Healthcare Copilot [97] (a ChatGPT-based system), provide clinicians and patients with reliable information on demand, making them instrumental for real-time medical inquiries and response applications. It integrates dialogue, memory, and processing components to facilitate safe patient-LLM interactions, enrich conversations with historical data, and provide summarized consultation reports. These systems can support healthcare professionals with structured and accurate information because they can rapidly integrate vast medical knowledge. This way, it is possible to ensure safe and accessible responses to frequently asked questions about various health conditions. However, before these models are widely deployed in real-world healthcare settings, significant barriers, such as addressing the risks of biased or inaccurate outputs, need to be surpassed to avoid improper medical advice or misdiagnoses.

Clinical reports, such as radiology reports, discharge summaries, and patient clinic letters, are also areas where LLMs can be determinant. These reports refer to standardized documentation that healthcare workers complete after each patient visit, which usually involves text generation/summarization and information retrieval. Systems such as ChatCAD [82] and ChatCAD+ [98] are designed to process radiological image data, generating detailed, human-readable diagnostic reports that align with established report formats used by radiologists. By structuring output according to diagnostic standards, these models enhance consistency and efficiency in documentation. ChatCAD+ further optimizes this process with the incorporation of templates derived from real-world radiological reports, ensuring the coherence

and readability of AI-generated content in clinical workflows.

While these benefits, along with the capability to produce more comprehensive and precise reports, are recognized in LLM-generated reports, there are still reported issues with, among others, lack of conciseness, hallucinations and literal interpretations of input, lacking the assumption-based reasoning that human doctors use [68]. Additionally, due to the specialized content and the generative nature of the task, it is challenging to evaluate these LLMs. Current evaluation methods mainly rely on lexical metrics, which can result in biased and inaccurate assessments of the reports' contextual information.

LLMs can enhance medical education in various ways, such as providing detailed explanations, aiding in language translation, answering questions, assisting with exam preparation, and offering Socratic-style tutoring. These applications can involve text generation, simplification, semantic similarity, and information retrieval. Rather than building models from scratch, leveraging the knowledge synthesis, question-answering, and content-generation capabilities of existing advanced models is often more efficient. An example of this is ChatGPT [99], which can offer explanations and clarifications on complex medical topics, supporting self-study and enhancing comprehension. Another relevant example is MedGemini [100], a multimodal model that can interpret medical images and create detailed reports, which is valuable for training diagnostic skills. However, a potential drawback of using LLMs in medical education is the lack of ethical training and inherent biases in training datasets, which, if left unaddressed, may permeate the generated content, reinforcing stereotypes and risking discriminatory outcomes in medical education [101].

Clinical coding, which includes systems like the International Classification of Diseases (ICD), medication codes, and procedure codes, is fundamental in healthcare for standardizing diagnostic, procedural, and treatment data. LLMs present a promising solution for automating clinical coding by extracting medical terms from clinical notes and assigning accurate codes, including ICD and medication codes, reducing healthcare professionals' workload and enhancing coding accuracy [102]. PLM-ICD [102], an LLM fine-tuned on the RoBERTa model, leverages medical-specific knowledge and performs well on large datasets like MIMIC-II and MIMIC-III. Similarly, DRG-LLaMA [103] uses LLaMA with parameter-efficient techniques to streamline coding. However, besides the already mentioned challenges of potential biases and hallucinations, and unlike traditional multi-label classifiers that constrain outputs to valid codes, generative LLMs can inadvertently generate non-existent or incorrect codes, especially with lengthy input texts. This issue highlights the need for proactive error detection mechanisms before integrating these models into electronic health records (EHRs) [104].

Medical language translation is another critical clinical area that serves two primary functions: translating medical terminology between languages and converting medical dialogue into everyday language for non-professional understanding, enhancing informed decision-making, fostering shared understanding and increasing patient satisfaction. Multilingual LLMs developed for medical translation are often fine-tuned on parallel corpora, using diverse datasets like scientific articles, clinical notes, and medical glossaries to capture the precise meanings of medical terms across languages. Models such as Medical mT5 [105], Apollo [106], and BiMediX [107] allow accurate translation across languages like En-

glish, French, Spanish, Chinese, and Arabic, supporting knowledge sharing across linguistic boundaries. When translating technical medical dialogue into everyday language, it is essential to fine-tune LLMs on datasets that include both technical conversations and their more basic explanations. Techniques such as retrieval augmentation, which pulls relevant everyday-language explanations from external sources, can improve clarity and accuracy in translated dialogue [108]. This application raises ethical concerns due to the potential discriminatory language inadvertently introduced by the LLM and other issues like the risk of misinterpretation and misinformation, which can lead to harmful consequences. Using high-quality, peer-reviewed medical sources in model training is crucial to help mitigate these risks.

Medical robotics is transforming healthcare, particularly in precision tasks like surgery and imaging. Integrating LLMs into these systems enhances decision-making, communication, and control. Notable systems demonstrate the potential of LLMs in robotics. SuFIA [109] leverages GPT-4 Turbo in robotic surgery for planning and task control. UltrasoundGPT [110] integrates LLMs with ultrasound robots for dynamic, prompt-based motion adjustments, improving scan efficiency. Brainlab's Loop-X [111], an X-ray-guided surgical robot, uses GPT-4 to enhance communication and precision in X-ray imaging. Integrating these models in medical robotics also raises safety issues, as LLMs may misinterpret human intent in shared environments, posing risks [112].

### 3.1.4 Ethical Implications of AI in Healthcare

Integrating AI into healthcare systems presents promising opportunities, however, with these advancements come pressing ethical implications that need to be addressed for safe and fair deployment. This chapter explores key ethical issues associated with AI in healthcare, focusing on privacy, bias, accountability, transparency, and the broader impact of AI dependence on healthcare practices [1].

#### 3.1.4.1 Privacy and Data Security

Medical data is inherently sensitive, including personal health records, diagnostic information, treatment histories, and even genetic data that could uniquely identify individuals. Handling and storing this type of data is highly regulated by privacy laws like HIPAA in the United States and GDPR in the European Union, and their handling by AI models, especially LLMs, raises significant privacy concerns. For instance, LLMs require vast amounts of data to function effectively, leading to the risk of unintentional data leakage, especially if personal health information is inadequately anonymized or accessed through malicious breaches.

Data security risks are compounded by the potential misuse of LLMs like ChatGPT where input data from users might be stored or inadvertently exposed to unauthorized parties. [113] discusses these risks, emphasizing that even anonymized data could be re-identified under certain conditions. This calls for robust data handling protocols and explicit consent frameworks for any use of LLMs in clinical environments.

#### 3.1.4.2 Bias and Fairness

Bias in LLMs poses a serious risk to equitable healthcare. Since LLMs are trained on vast datasets that may reflect existing societal and cultural biases, they can inadvertently perpetuate disparities in healthcare delivery. For instance, diagnostic models might underperform for minority populations if the training data lacked sufficient representation of these groups. [114] notes that addressing bias is critical to prevent unequal treatment outcomes and ensure that AI-driven healthcare tools do not disproportionately favour one demographic over another.

Moreover, biases may manifest in the language used by LLMs when interacting with patients. As [114] points out, LLMs can produce responses that reflect harmful stereotypes or reinforce existing healthcare inequalities. To mitigate these issues, LLMs must undergo rigorous bias evaluation and adjustment during development, with an emphasis on inclusive training datasets and continual auditing for equitable performance across diverse patient groups.

#### 3.1.4.3 Accountability and Transparency

The deployment of AI in healthcare raises fundamental questions about accountability and transparency. If an AI system provides incorrect or harmful recommendations, it remains unclear whether responsibility lies with the developers, healthcare providers, or the AI system itself. [115] suggested that accountability frameworks are needed to clearly delineate roles and responsibilities when AI-driven errors occur in clinical settings.

Transparency is equally important for patient trust and informed consent. Patients and healthcare providers should understand how an AI system arrives at its recommendations, which is a challenge for the "black box" nature of many LLMs. [113] highlights the opacity of models like ChatGPT, which can be especially problematic in high-stakes healthcare environments. Efforts to improve transparency may include explainable AI techniques that enable LLMs to provide interpretable reasoning for their outputs. Without transparency, it is challenging for patients to trust AI-assisted healthcare, and for clinicians to use AI tools responsibly.

#### 3.1.4.4 AI Dependence and the Impact on Healthcare Practices

The growing reliance on AI in healthcare brings risks associated with over-dependence on automated systems. [113] raises concerns about AI replacing critical thinking, especially if clinicians overly rely on AI recommendations without cross-verifying with clinical judgment, leading to declining diagnostic skills and critical reasoning among healthcare professionals.

Furthermore, AI tools could inadvertently influence the patient-clinician relationship. If LLMs like ChatGPT handle initial patient interactions, patients may perceive AI as a substitute for personalized medical advice, potentially undermining the patient-centred healthcare model and impacting the emotional support system, which is vital to patient care. Therefore, healthcare providers must balance AI assistance with human oversight to preserve quality in patient interactions and clinical practices [1].

The ethical challenges posed by AI in healthcare require comprehensive legal and ethical safeguards. Current laws often lag behind AI advancements, and new frameworks are needed to address specific AI-related issues, such as data re-identification risks, accountability in automated decision-making, and bias prevention.

## 3.2 Retrieval-Augmented Generation in Healthcare

RAG has emerged as a transformative approach in healthcare applications, enhancing LLMs by integrating up-to-date medical knowledge into their responses. This section explores techniques used with RAG to improve LLMs performance in healthcare.

Some common challenges faced by LLMs are related to generating factually inaccurate content, often referred to as hallucinations, which occur when models create responses that seem plausible but lack factual grounding. This issue becomes particularly troubling in areas requiring accurate and current data, such as healthcare, where errors can have serious consequences. Additionally, traditional LLMs can struggle with transparency in reasoning, making it difficult to trace the source of information and evaluate its reliability. The RAG framework has emerged as a promising solution to these issues by introducing an architecture that combines LLMs with retrieval-based mechanisms, drawing on relevant knowledge from external sources. Through retrieval, RAG can access and integrate external, contextually relevant knowledge that updates continuously, enhancing accuracy and credibility in responses and allowing it to go beyond the fixed knowledge base of a pre-trained LLM, drawing on current or specialized datasets relevant to the user's query. Consequently, RAG-equipped LLMs are more suitable for real-world applications where up-to-date and domain-specific knowledge is essential, and they reduce reliance on the static knowledge locked within LLM parameters [116].

[117] discussed the concept of RAG for enhancing LLMs in handling knowledge-intensive tasks. The authors present two approaches for RAG: RAG-Sequence and RAG-Token. RAG-Sequence uses the same retrieved passage throughout the generation process, while RAG-Token allows different retrieved passages for each token generated. The study demonstrates RAG's effectiveness across various tasks, including open-domain question answering, abstractive question answering, fact verification, and even Jeopardy-style question generation. The model achieves state-of-the-art performance in open-domain QA, showing that RAG can produce more diverse, specific, and factual language than parametric-only models. Additionally, the paper explores the benefits of RAG's non-parametric memory, such as the ability to update the knowledge base dynamically without retraining, which addresses the limitations of LLMs that rely solely on static, parametric memory.

[118] evaluated how RAG could enhance LLMs for knowledge-intensive tasks in nephrology by addressing common issues like accuracy and relevance in clinical settings. By integrating RAG, the study aimed to create a ChatGPT model that could dynamically retrieve and incorporate real-time, authoritative information aligned with KDIGO 2023 guidelines for chronic kidney disease (CKD). The

results demonstrated that the RAG-enhanced ChatGPT significantly improved the precision and relevance of responses. For instance, it provided treatment recommendations that adhered closely to the latest CKD protocols, such as accurately suggesting SGLT-2 inhibitors for diabetic CKD patients and tolvaptan for ADPKD, which the standard model missed. Furthermore, in multidisciplinary cases like diabetic nephropathy, the RAG model successfully integrated insights across fields, supporting comprehensive treatment strategies. Regular updates from sources like PubMed ensured the model remained current without retraining, highlighting RAG's ability to maintain relevance in rapidly evolving medical domains.

[119] presents a benchmark designed to evaluate RAG systems specifically for medical applications and handles the limitations of LLMs in medicine, such as hallucinations and outdated knowledge, by employing RAG to retrieve relevant information from external sources. One of the author's main contributions is the introduction of MIRAGE, a benchmark containing 7,663 questions across five medical QA datasets, and evaluating RAG performance using the MEDRAG toolkit, which integrates multiple medical corpora, retrieval methods, and LLMs. The results showed that RAG can significantly enhance the accuracy of LLMs in medical contexts, improving performance by up to 18% on standard prompting techniques, like chain-of-thought prompting. With RAG enhancements, GPT-3.5 and Mixtral models achieved accuracy levels comparable to GPT-4, highlighting RAG's potential to elevate model performance cost-effectively. The study highlighted a log-linear scaling relationship where more retrieved snippets correlated with improved accuracy up to a point, beyond which adding snippets introduced noise and also confirmed a "lost-in-the-middle" effect: accuracy dropped when the most relevant snippet was positioned in the middle of the retrieval list rather than at the beginning or end, which hints at the potential for better snippet arrangement strategies for future RAG implementations.

[120] investigated the effectiveness of using Llama 2 RAG techniques to process EHRs in aged care, focusing on two main tasks: generating structured summaries of clinical notes about malnutrition for residents and extracting malnutrition risk factors from nursing notes in real-world EHR data from Australian aged care facilities. They employed zero-shot learning with Llama 2 on 2,278 labelled notes, comparing performance with and without RAG. RAG was used to retrieve specific, relevant information from a large dataset, enhancing the model's ability to produce accurate summaries and risk factor identifications without extensive retraining. Llama 2 achieved an accuracy of 93.25% without RAG but showed occasional hallucinations, such as inserting incorrect ages or weights. When RAG was integrated, accuracy increased to 99.25%, significantly reducing errors and aligning more closely with the standard summaries created by experts. For the second task, identifying malnutrition risk factors, the model achieved 95% accuracy on notes with explicit risk factors and 85% on notes lacking specific details, where it occasionally inferred incorrect factors (extrinsic hallucination). RAG did not further improve accuracy in this task, as the retrieval-based approach is less relevant for notes already containing risk information.

[121] presented Health-LLM, an innovative healthcare model that combines LLMs with RAG to enhance personalized disease prediction and health management, integrating patient health reports with

external medical knowledge to predict diseases and generate personalized health advice. Health-LLM operates by extracting health-related features from patient data, assigning scores to these features using the Llama Index (a method designed to enhance LLM responses with precise medical knowledge), and applying an XGBoost classifier for disease prediction. The study demonstrates that Health-LLM outperforms existing approaches in terms of accuracy and F1 scores, achieving 83.3% accuracy and an F1 score of 0.762, surpassing models like GPT-4 and fine-tuned LLaMA-2 in various clinical prediction tasks. Through its integration of RAG, Health-LLM enables dynamic updates, making it adaptable to individual patient needs and continuously improving prediction accuracy by adding or refining features.

[122] presented GastroBot, a chatbot designed to provide accurate, contextually relevant information about gastrointestinal diseases using RAG. GastroBot is tailored for the Chinese medical field and utilizes RAG to enhance the precision and relevance of answers by pulling information from a specialized database of 25 clinical guidelines and 40 recent articles on gastrointestinal conditions. It uses a fine-tuned embedding model (based on Alibaba's GTE model) specifically trained on gastrointestinal knowledge, resulting in an 18% improvement in hit rate over the base model and a 20% improvement over OpenAI's embedding model. The RAG system for GastroBot achieved high scores in RAGAS metrics, with a context recall rate of 95%, faithfulness of 93.73%, and answer relevance of 92.28%. GastroBot outperformed baseline models (Llama 2, ChatGLM-6B, and Qwen-7B) in providing accurate, safe, and user-friendly responses. In human evaluations, it also received high scores for safety, usability, and smoothness, indicating its effectiveness in clinical applications.

[123] introduced RankRAG, a novel framework that integrates context ranking with RAG within a single LLM. RankRAG aimed to enhance LLM performance on knowledge-intensive tasks by simultaneously instructing the model to rank relevant contexts and generate answers. Traditionally, RAG systems use a separate ranking model to identify the top-k relevant contexts from a larger set of retrieved documents, which are then fed to the LLM for answer generation. RankRAG, however, combines these two functions—context ranking and answer generation—within the same instruction-tuned model, leveraging a small amount of ranking data to improve the system's ability to filter out irrelevant content. The model's two-stage instruction tuning significantly improved accuracy over standalone RAG models across nine general-domain and five biomedical benchmarks, outperforming models like GPT-4 and ChatQA-1.5 (a top RAG baseline) on tasks requiring high recall and precise context selection. The RankRAG framework showed promising adaptability, especially for specialized tasks like biomedical information retrieval, without needing domain-specific fine-tuning, marking a notable advance in RAG system efficiency and generalization.

[124] explored methods to optimize RAG for personalized LLMs according to individual user preferences. The paper introduces two primary optimization algorithms aimed at refining the retrieval process: one based on reinforcement learning, where the retrieval model is rewarded for selecting documents that lead to effective personalized responses, and another based on knowledge distillation, aligning retrieval relevance with the LLM's downstream performance. Additionally, the authors propose a retrieval selection model that dynamically chooses between different retrievers (e.g., recency-based, se-

mantic matching) based on the specific input, enhancing personalization flexibility. Evaluated using the LaMP benchmark for personalization tasks, the methods significantly improved performance across six out of seven tasks, advancing state-of-the-art personalization techniques in LLMs.

RAG's ability to dynamically incorporate specialized medical information significantly improves LLM performance in healthcare, supporting accurate clinical decision-making, patient education, and personalized health recommendations. As research continues to refine retrieval and ranking mechanisms, RAG's role in healthcare applications is likely to expand, offering increasingly reliable, context-aware support for medical professionals and patients alike.

## 3.3 Evaluation Metrics in LLMs

This section explores the metrics used to evaluate large language models, the benchmarking strategies and the datasets.

### 3.3.1 Performance Metrics

With the increasing popularity of chatbots, it is important to develop methods to evaluate their performance.

Lexical evaluation compares the surface-level text using metrics like accuracy, EM, precision, recall, and F1 scores. Accuracy measures how often the system's answer is correct, while EM checks if the system's output matches the reference answer exactly. Precision and recall evaluate the proportion of correct words in the system's response and how well it covers the reference answer, respectively, with the F1 score balancing these two measures. In healthcare, these metrics are applied for structured tasks like diagnostic predictions [2].

Additional lexical metrics like ROUGE, BLEU, and METEOR are commonly used in tasks like text generation and machine translation. These metrics assess word overlap and account for synonym matching or phrasing differences. BLEU is traditionally used in translation tasks, focusing on precision by measuring the overlap of n-grams between model outputs and reference texts. Similarly, ROUGE is often applied to summarization tasks, evaluating how many words or n-grams in the generated text match those in reference summaries. Human Equivalence Score (HEQ) is another metric that compares the system's answers to human performance, and VQA Accuracy evaluates how well the system aligns with answers given by human annotators.

Semantic evaluation, on the other hand, moves beyond surface-level matching to focus on the meaning of the text. Word Mover's Distance (WMD) and Sentence Mover's Similarity (SMS) are metrics that calculate how far the words or sentences need to "travel" in meaning space to match the reference. BERTScore leverages BERT embeddings to measure the cosine similarity between predicted and reference answers, while BLEURT and MoverScore use advanced embeddings to model semantic similarity. BARTScore measures the probability of generated text based on its relevance and fluency. Semantic

Answer Similarity (SAS) is a metric specifically designed to evaluate the meaning of generated answers by comparing their contextual similarities.

Lastly, LLM-based evaluation methods leverage the capabilities of large language models themselves to judge the quality of generated text. Metrics like GPTScore use pre-trained LLMs such as GPT-3.5 to assess the quality of responses based on internal knowledge. Others, like G-Eval, employ chain-of-thought reasoning, allowing LLMs to systematically evaluate responses. These LLM-based methods are particularly useful in scenarios where human-like judgment is required to assess factual correctness and the fluency, coherence, and relevance of the answers.

In the biomedical domain, specialized metrics like CheXbert and RadGraph were introduced to account for the unique requirements of clinical applications. The first evaluates radiology reports by measuring similarity with standard medical terms, allowing for more precise assessment in medical imaging contexts. The latter serves a similar purpose, evaluating structured information extracted from radiological texts and ensuring that generated medical reports align accurately with clinical findings [125].

[126] explored using ChatGPT as an evaluation metric for assessing natural NLG model outputs. The author aims to evaluate whether ChatGPT can be a reliable substitute for human evaluation in tasks like summarization, story generation, and data-to-text generation. The ability to score model outputs similarly to human judgments was assessed by designing task-specific and aspect-specific prompts, such as those focusing on coherence, relevance, consistency, and fluency. It was observed that ChatGPT highly correlates with human evaluations, particularly in creative tasks like story generation, where multiple valid outputs may exist. However, the study also highlights ChatGPT's sensitivity to prompt design, noting that well-structured prompts are essential for accurate evaluation.

[100] used several evaluation metrics to assess the performance of Med-Gemini models across a variety of medical tasks. Accuracy was used for Medical Question Answering tasks, particularly with the MedQA benchmark, to measure the proportion of correctly answered multiple-choice questions. F1 Score was applied in tasks like entity recognition and information extraction. BLEU and ROUGE were used in text generation tasks, such as summarization and referral letter generation, to compare the generated text with reference texts based on n-gram overlap. MRR was used for information retrieval tasks, such as Needle-in-a-Haystack retrieval from extensive medical records, measuring the rank at which the correct answer appears. Image Classification Accuracy and MAP evaluate the model's visual understanding capabilities in image classification and visual question-answering tasks. Finally, Human Evaluation Score was used in tasks like medical text summarization and referral letter generation to assess the quality of generated content against clinical standards, ensuring the outputs are both accurate and practically useful in medical settings.

In [90], accuracy was used to evaluate the model's ability to answer questions rigorously, focusing on whether the answers align with expert-provided answers. For more complex questions requiring reasoning, MEDITRON's performance was also evaluated with CoT prompting, which encouraged step-by-step reasoning, assessing how well the model followed logical steps in reasoning, a critical factor for

medical decision-making tasks. SC-CoT measured the model's consistency when multiple reasoning paths were followed and accurately evaluated MEDITRON's ability to consistently produce reliable answers under varied reasoning prompts by generating several reasoning sequences and selecting the most consistent answer.

### 3.3.2 Cost Metrics

The deployment and use of LLMs involve substantial costs, primarily in inference and operational expenses.

The first, which accounts for the computational power required to generate model responses, is particularly high for large-scale models, considering that deploying models like GPT-4 for customer support applications can cost around $21,000 monthly for small businesses [127].

The latter, operational, also differs significantly between deployment types. Local models, such as LLaMA2-70B, require substantial hardware and maintenance, often exceeding $5,200 monthly when hosted on cloud GPU instances, which underscores the initial and ongoing investment required [128]. On the other hand, service-based models available through APIs have a different cost structure. These models generally charge per token, with fees tied to both prompt (input) and generation (output) lengths and, sometimes, a fixed cost per query. Within these types of models, the pricing varies significantly: OpenAI's GPT-4 costs around $30 per million input tokens, but Textsynth's GPT-J, a more affordable option, costs only $0.20 for the same input size [129].

In some cases, additional costs arise from monitoring and misuse prevention, especially in API-accessible models. Misuse detection can increase operational expenses if users must employ evasion techniques like IP rotation due to restrictions or penalties enforced by providers [130]. Some applications also benefit from a human-machine team setup, where human review complements automated responses. While often yielding high-quality outputs, this method still incurs personnel costs and adds complexity compared to fully automated approaches.

The increasing adoption of LLMs in diverse applications requires a comprehensive understanding of the associated costs and metrics to facilitate cost-effective usage.

#### 3.3.2.1 Comparison of Local and Service-Based LLMs

Local models are potentially cost-effective for applications with high query volumes, where the one-time higher infrastructure cost caused by the need for specialized hardware and ongoing maintenance can be offset by avoiding per-query fees. Furthermore, local models provide greater control over data processing, which enhances data privacy and regulatory compliance [130] [128].

In contrast, service-based models provide a more flexible pay-as-you-go pricing structure, mitigating the initial investment barrier. It is advantageous for applications with fluctuating demand, as users only pay for actual usage, though it can become expensive with high query volumes. Additionally,

data processed through service-based models is typically handled on external servers, raising potential privacy concerns. However, these models also have built-in security and misuse monitoring features, which can add value, especially for sensitive applications [131].

[132] provided an extensive cost analysis comparing the deployment of local LLMs via the LlamaDuo pipeline to using token-based API access from cloud-based LLM services like GPT-4. The author's analysis covers initial fine-tuning and operational deployment costs under different workloads. The fine-tuning of local LLMs involves a one-time initial setup, highlighting that fine-tuning the Gemma 7B model on synthetic data requires about $50 per hour on 8x A100 GPUs, which amounts to an estimated total of $800 when accounting for multiple cycles and hyperparameter optimization. The author also provides costs for running local models under both light and heavy workloads for operational deployment, which are consistent and predictable, providing a stable budget for organizations. A light workload scenario, requiring only a single instance of the local model, runs on an L4 GPU at $2,539 monthly, with $30 added for electricity. In contrast, a heavy workload needing multiple instances (8x L4 GPUs across different servers) costs about $20,312 per month, plus $240 for electricity. Comparing these costs with those for service LLMs, like GPT-4o, reveals significant differences. Service LLM costs depend on usage: a light workload using 10 million input tokens and 1 million output tokens daily would cost around $1,950 per month. In comparison, a heavy workload with 50 million input and 5 million output tokens daily would cost $9,750 monthly, which shows that cloud services are more expensive over time, especially in high-traffic scenarios. The author highlights the long-term economic benefits of local LLMs, where initial fine-tuning and setup are one-time costs, in contrast to the ongoing expenses of cloud-based API services. Deploying a local model under heavy workloads becomes cost-effective within two months. After a year, costs for cloud LLMs reach $117,000 compared to just $23,992 for local deployment—making the local model nearly five times cheaper. Local LLMs also provide predictable, fixed costs, avoiding variable token-based fees and enabling better budget forecasting. They offer complete control over performance tuning without extra API costs, support data privacy by reducing third-party exposure, and mitigate the need for additional privacy measures.

[133] discusses the financial challenges of deploying massive, service-based LLMs, which involve high initial training costs, substantial ongoing usage fees, and complex infrastructure requirements. It emphasizes the high costs associated with models like GPT-4, which require considerable computational resources and are financially out of reach for many institutions due to hardware demands and proprietary model access costs. The authors explore using local LLMs—smaller, open-source models like LLaMA— that can be fine-tuned on specific datasets and optimized for structured output while running on more accessible hardware, aiming to balance cost-efficiency with performance by leveraging techniques like quantization to reduce memory requirements and potentially allow for CPU-based inference, minimizing the infrastructure costs associated with GPU deployments. This work demonstrates that local LLMs, while not fully matching the scope of larger models, can deliver cost-effective, high-quality results for specific, structured data tasks, such as extracting condition codes from medical reports, without incurring the substantial and continuous costs of service-based LLMs.

#### 3.3.2.2 Cost Optimization Strategies

To address the diverse costs, several optimization strategies have been explored.

Prompt adaptation is a straightforward technique to reduce costs by minimizing the length of prompts sent to the model, effectively lowering token usage without compromising the quality of outputs [129]. In addition to prompt efficiency, LLM approximation aims to use simpler models or cached responses for predictable queries, thus avoiding the need to invoke more expensive models continuously. For instance, cached answers from high-confidence queries can be reused for subsequent similar queries, reducing the need for repeated computation and consequently saving costs [131].

An impactful approach is the LLM cascade method, introduced in frameworks like FrugalGPT, where queries are sequentially passed through a series of models with ascending cost and accuracy. Less complex queries are handled by low-cost models, while more challenging ones move up the cascade to models with higher accuracy (and cost), such as GPT-4. This technique has demonstrated up to 98% cost savings by dynamically adapting the model selection to query complexity [129].

[132] outlines a cost optimization strategy that transitions from service-based LLMs to locally managed, fine-tuned LLMs using the LlamaDuo pipeline. This approach minimizes expenses by replacing ongoing token-based API fees with one-time fine-tuning costs and fixed hardware investments. By generating synthetic data for iterative training, LlamaDuo allows smaller models to achieve performance comparable to larger, cloud-based models, targeting specific tasks without incurring high data acquisition or API costs. This strategy offers scalable deployment on local hardware, providing predictable expenses and enhanced control, making it a practical, cost-effective solution for organizations aiming to reduce long-term AI operational costs.

[134] introduced QC-Opt, a framework designed to reduce the costs of using LLMs for tasks like summarization and question answering. QC-Opt selects the most cost-effective LLM based on quality, budget, and latency needs without compromising output quality. It includes a quality prediction model to estimate output quality without running the LLM, an algorithm for optimal model selection, and a token optimization tool to reduce input length, thereby cutting costs. QC-Opt achieves significant cost savings (40-90%) while maintaining or slightly improving output quality across various datasets.

#### 3.3.2.3 API Pricing for Popular LLMs

The cost of using LLM APIs varies based on the provider and the model's capabilities. As an example, OpenAI's GPT-4 with a 128K context window costs approximately $5 per million input tokens and $15 per million output tokens, while their GPT-4 Turbo model, also with a 128K context window, costs around $1 and $3 per million tokens for input and output, respectively. Anthropic's Claude 3.5 Sonnet, with a 200K context window, has a higher output cost at about $15 per million tokens but a similar input

cost to OpenAI's Turbo variant at $3 per million tokens. Google's Gemini 1.0 Pro offers a relatively lower cost per million tokens at $0.50 for input and $1.50 for output but has a smaller context window of 32K. Meta's Llama 3.1 405B, another alternative, charges about $1.79 per million tokens for both input and output, maintaining a 128K context window. Table 3.3 provides an overview of API pricing for popular LLMs. While these prices are examples and subject to change, they highlight the diversity in pricing models across providers, encouraging users to refer to official sources for the most current information.

| Model Name | Provider | Input Cost per Million Tokens | Output Cost per Million Tokens | Context Window |
|---|---|---|---|---|
| GPT-4 | OpenAI | $5.00 | $15.00 | 128K |
| GPT-4 Turbo | OpenAI | $1.00 | $3.00 | 128K |
| Claude 3.5 | Anthropic | $3.00 | $15.00 | 200K |
| Gemini 1.0 Pro | Google AI | $0.50 | $1.50 | 32K |
| Llama 3.1 405B | Meta | $1.79 | $1.79 | 128K |

Table 3.3: API Pricing for Popular LLMs

### 3.3.3 Energy Consumption Metrics

Recent research offers a range of practical metrics to help quantify and manage energy use effectively. These metrics provide insights into the efficiency of LLMs during training and inference, enabling targeted optimizations.

Energy per Token measures the energy consumed for each token generated or processed by the model. This fine-grained metric is particularly useful in comparing the energy efficiency of models with varying sizes and configurations under similar workloads. Another comprehensive measure is Total Energy (kWh), capturing the total kilowatt-hours consumed throughout training or inference. This metric is ideal for comparing multiple runs or model configurations, especially when assessing overall project energy costs.

A data centre-specific metric, Power Usage Effectiveness (PUE), calculates the ratio of total facility energy (including cooling) to IT equipment energy, offering insights into facility efficiency. Lower PUE values signal better efficiency, guiding decisions on when and where to deploy models for minimized energy costs associated with infrastructure. GPU Power Capping is another valuable metric; by setting limits on GPU power draw (e.g., 150W instead of 250W), energy consumption and processing time can be measured to identify optimal power caps that balance energy savings with performance.

Energy per Request quantifies the energy used for processing a single inference request, providing a measure for efficiency in production environments. Similarly, Energy-Efficiency of Batch Size assesses how energy consumption varies with different batch sizes. Larger batches may reduce energy per request but could increase latency, so this metric helps identify a balance for production tasks.

At a computational level, Normalized Energy Consumption (Joules per FLOP) evaluates energy usage per floating-point operation, facilitating comparisons across model architectures or optimizations, such as quantization. GPU Frequency Scaling Impact is also useful, as it tracks energy consumption at

different GPU frequencies; adjusting GPU frequencies can save energy, although it may affect performance, making this metric crucial for tuning energy-performance trade-offs.

Other important metrics include Daily/Seasonal Energy Efficiency Variations, which tracks energy efficiency changes throughout the day or season. Running tasks during cooler periods can reduce cooling costs in facilities with fluctuating cooling needs. Finally, Carbon Intensity of Energy Source captures the carbon emissions associated with energy usage (measured in grams of CO2 per kWh), allowing scheduling of tasks during times of higher renewable energy availability, reducing the carbon footprint.

[135] investigated the energy consumption of running LLMs during the inference stage, focusing on Meta's LLaMA model as a case study. It assessed how different LLM sizes (from 7 billion to 65 billion parameters) perform regarding energy and computational costs across various hardware setups, including different GPU configurations. Through experimental benchmarks on GPUs like NVIDIA's V100 and A100, the study provided a detailed analysis of energy metrics such as energy usage across different sizes of Meta's LLaMA model (7B, 13B, 65B) using NVIDIA V100 and A100 GPUs. Larger models, like LLaMA 65B, consumed significantly more energy, with an average of 3-4 Joules per decoded token. While A100 GPUs increased throughput compared to V100s, they also drew more power, especially in distributed multi-GPU setups. Scaling batch sizes and sharding (up to 32 GPUs) boosted throughput but at a high wattage cost. Power capping on A100s (reducing power from 250W to 175W) achieved a 23% energy saving, although further reductions yielded diminishing returns. The study highlights that GPU memory utilization was often low (20-25%), suggesting the potential for multiple workloads on the same hardware to improve energy efficiency. These findings emphasize that strategic configuration and energy-saving techniques can mitigate the environmental impact of large-scale LLM deployments.

Every interaction with a LLM incurs an energy cost during inference, along with the energy required for the model's initial training and preparation for production. [136] explored the energy costs of operating LLMs and compared these to the energy expenditure involved in similar human activities. The initial training of LLMs requires vast computational resources, leading to a substantial energy footprint. Training large models, such as GPT-3 or GPT-4, can require thousands of GPU hours, resulting in substantial kWh consumption—often on the scale of megawatt-hours. This training is a one-time but intensive energy investment. When compared to the energy required by a human using a computer for the same task, the results show that LLMs can, in some cases, be less efficient, especially given the training overhead. However, LLMs might save time and resources in high-repetition or data-intensive tasks, where human efforts would be inefficient. This study suggests that, while LLMs might increase efficiency by speeding up information processing, these gains come at an environmental and energy cost that must be carefully considered.

[137] explored methods to improve the energy efficiency of LLMs in data centres, where energy use has become a major challenge due to the intensive requirements of LLMs. It finded that input and output length heavily influence energy use, with longer inputs increasing compute load during prefill phases and longer outputs affecting decode phases. Lowering GPU frequency proves to be an effective energy-saving strategy under light or medium workloads, reducing power by about 20% without sacrificing

performance. Additionally, increasing parallelism across GPUs improves performance but raises energy demands due to communication overhead, with smaller configurations often being more efficient for low-throughput needs. Batch size also plays a critical role: while larger batches boost throughput, they increase energy costs and may miss latency targets, highlighting that batch size optimization is crucial. The study advocates for adaptive, workload-aware configurations, including frequency tuning and dynamic batching, to balance energy use and performance, supporting more sustainable LLM deployment in data centres.

[138] provided a comprehensive analysis of techniques that can decrease energy demands without compromising model performance. Approaches discussed include power-capping GPUs to limit their energy use, optimizing data centre scheduling to leverage periods of higher energy efficiency, and considering seasonal and daily temperature variations that impact cooling neexds. The study highlights that simple adjustments, like reducing maximum power to 150W on GPUs, can reduce energy consumption by up to 15% with minimal impact on training times. Additionally, the paper advocates for energy-aware practices, such as shortening training duration when only marginal performance gains are achieved and using efficient, climate-conscious data centres.

### 3.3.4 Benchmarking Datasets

In the literature, multiple-choice question answering, with accuracy as the main metric, is the primary method for assessing the performance of medical language models. Following this standard, we present three prominent medical QA benchmarks for evaluation.

#### 3.3.4.1 MedQA-USMLE

MedQA is a comprehensive dataset proposed by [139] that focus on evaluating AI models' abilities to answer challenging multiple-choice medical questions. These questions are sourced from professional medical board exams, such as the United States Medical Licensing Examination (USMLE), and are crafted to reflect real-world clinical decision-making. The dataset includes questions in English, simplified Chinese, and traditional Chinese, making it suitable for multilingual applications in medical AI. Each question presents four possible answer choices, encouraging models to engage in complex reasoning and knowledge retrieval to determine the correct answer.

The MedQA dataset comprises over 60,000 questions, each paired with extensive supporting content from medical textbooks and resources. This document collection serves as background knowledge, enabling open-domain question-answering models to retrieve relevant information as they search for accurate responses. The dataset's questions span various specialities, from pharmacology and surgery to anatomy and physiology, presenting a broad and realistic challenge for models in clinical and biomedical contexts.

MedQA is structured to evaluate not only factual recall but also diagnostic and inferential reasoning, as questions often involve complex clinical scenarios requiring multi-hop reasoning. This makes

MedQA particularly challenging compared to simpler question-answering datasets. As a key resource in biomedical AI, MedQA provides a robust benchmark for developing QA systems to support medical professionals by delivering precise, evidence-based answers in clinical practice and research settings.

#### 3.3.4.2 MedMCQA

The MedMCQA dataset is a large-scale, multiple-choice question-answering (MCQA) dataset for the medical domain. It was designed by [140] to evaluate AI models on complex medical questions derived from real-world medical entrance exams (AIIMS and NEET-PG) in India. The dataset consists of over 194,000 questions spanning 21 medical subjects, covering 2,400 healthcare topics. Each question has four answer options, one or more correct answers, and an explanation that provides reasoning for the correct choice.

MedMCQA emphasizes diversity and complexity by incorporating questions that require advanced language comprehension and reasoning skills across a broad range of medical subjects, such as pharmacology, pathology, and surgery. The questions test more than ten types of reasoning abilities, including causal reasoning, comparison, and diagnostic skills. MedMCQA's large size and diversity make it a valuable benchmark for developing and evaluating QA systems that can assist with medical knowledge.

For evaluation, MedMCQA is split into train, development, and test sets using an exam-based split to mimic real-world medical exam scenarios and avoid overlap of similar questions. The baseline performance of current state-of-the-art models (such as PubMedBERT and BioBERT) on this dataset highlights its difficulty, with even the best models performing significantly below human-level accuracy.

#### 3.3.4.3 PubMedQA

PubMedQA is a biomedical question-answering dataset specifically designed by [141] to test models' abilities to answer research-oriented questions in the medical domain. Each instance in the dataset comprises a question derived from the title of a PubMed article, which typically reflects a study's primary research question. Alongside the question, PubMedQA provides a supporting abstract (excluding its conclusion) and a simplified yes/no/maybe answer based on the study's findings. This dataset is particularly valuable in assessing the reasoning required to interpret scientific abstracts, which are often dense with medical terminology and evidence.

The dataset is divided into three parts: PQA-L, PQA-U, and PQA-A. PQA-L consists of 1,000 expert-labeled examples intended for model testing and validation. PQA-U, with 61,200 unlabeled entries, is used to support semi-supervised training approaches. PQA-A is the largest subset, containing 211,300 examples generated heuristically to train models with an approximate domain understanding. This structured approach to data curation supports a range of model training and testing configurations, encouraging advancements in biomedical NLP and question-answering tasks.

PubMedQA's unique question design emphasizes causal relationships and treatment effects, which are common in medical research and often require multi-step reasoning and contextual understanding.

Evaluation metrics for the dataset focus on accuracy and F1 scores, measured in settings that either require or do not require contextual reasoning over the abstract.

## 3.4 Identified Research Gaps

### 3.4.1 Limited Exploration of RAG with Smaller Local LLMs in Healthcare

While RAG has shown significant promise in enhancing the performance of LLMs in healthcare, the current research predominantly focuses on large, service-based models like GPT-4 and Med-PaLM 2. Studies such as [118] and [119] have demonstrated how RAG can improve accuracy and reduce hallucinations in these extensive models by integrating up-to-date medical knowledge. However, there is a noticeable gap in exploring the integration of RAG with smaller, local LLMs. Smaller models like Llama 2 and Mistral offer advantages in terms of cost, energy efficiency, and data privacy, especially when deployed locally. Despite these benefits, research on leveraging RAG to enhance the capabilities of small local LLMs in healthcare remains limited. This gap highlights the need for studies investigating how RAG can be effectively combined with smaller models to achieve performance comparable to larger counterparts, thus providing a viable and resource-efficient alternative in clinical settings.

### 3.4.2 Cost and Energy Efficiency Analysis

The literature acknowledges the substantial costs and energy consumption associated with deploying large LLMs, particularly service-based models. Works like those by [132] and [133] discuss the financial and infrastructural challenges posed by models like GPT-4, emphasizing the need for cost-effective solutions. While some research compares local and service-based models, there is insufficient analysis focusing on how smaller local LLMs enhanced with RAG can reduce monetary and energy costs compared to larger models. Existing studies often overlook the potential savings in operational expenses and energy consumption that small local models might offer when integrated with RAG. This represents a critical gap, as understanding these cost-performance trade-offs is essential for organizations aiming to implement AI solutions in healthcare without incurring prohibitive expenses. Addressing this gap could demonstrate how smaller local LLMs with RAG not only preserve data privacy but also offer sustainable and economical advantages.

# Chapter 4

# Implementation

This chapter details the design and development of a pipeline to streamline the usage of RAG with a selected number of local models, along with the process of evaluation of key metrics and monetary and energetic costs of both local and service LLMs.

## 4.1 Solution Overview

Figure 4.1 illustrates the solution pipeline used, designed to enhance the quality and relevance of responses produced by a LLM using RAG, and evaluate the performance of service and local LLMs. In the first phase, an embedding model processes both the documents and an incoming query to generate vector representations (embeddings). These embeddings are then stored and indexed in a Vector Database, enabling efficient similarity searches. When a query arrives, the system retrieves the *top-k* most relevant document fragments from the Vector Database, which are then further refined by a reranker into the *top-x* segments. This subset of highly pertinent context is finally concatenated with the original prompt provided to the LLM.

Within the LLM component, few-shot learning techniques may be employed to supply examples or instructions that guide the model's reasoning process (CoT). The model's output is subsequently passed to an evaluator module, responsible for assessing the quality. By integrating retrieval and reranking steps with an advanced LLM, this architecture aims to deliver domain-specific and context-rich answers, ideally suited to complex information retrieval tasks that require precision and explainability. Important to highlight that during the work in this study the focus was on varying RAG parameters (*top-k* and *top-x* contexts), and everything after the contexts retrieved: prompt selection, LLM integration, and evaluation process (including Chain of Thought) to assess the performance of both local and service LLM in our selected dataset.

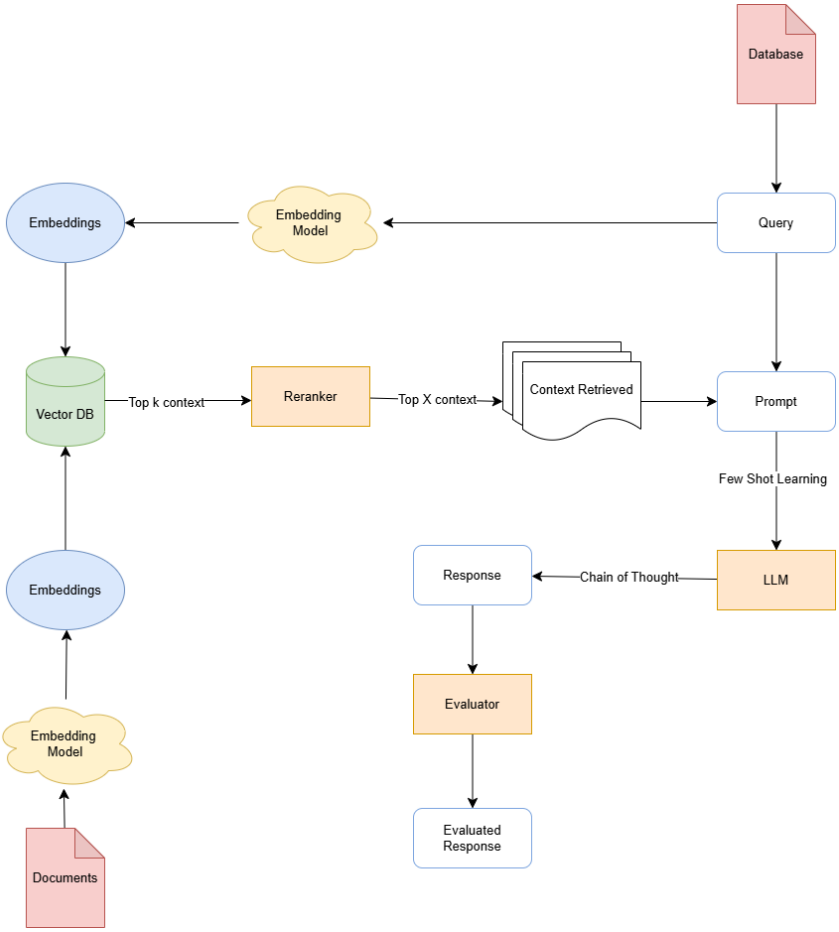


Figure 4.1: System Architecture

## 4.2 Prompting Strategies

The design of prompting strategies was essential to maximize the potential of LLMs. To evaluate the performance of the models, tests with multiple-choice and open-answer formats were performed. Two primary strategies, CoT prompting and Few-Shot Learning, were employed to address the diverse requirements of these tasks.

For QA tasks involving multiple-choice questions, a straightforward approach was employed to streamline the evaluation. The prompt explicitly instructed the model to provide a precise answer referencing only the letter of the chosen option. This method facilitated direct comparison between the model's output and the correct answer, ensuring efficiency and eliminating ambiguity during evaluation. CoT prompting was not applied to these tasks, as the focus was on achieving clarity and precision in selecting the correct choice without intermediate reasoning steps.

For open-ended tasks, Chain-of-Thought prompting was used to elicit reasoning capabilities in the models. This approach enabled the generation of intermediate reasoning steps, helping the model break down complex problems into smaller, logical sub-components. Studies such as [17] demonstrated the

effectiveness of CoT in improving performance on multi-step tasks and this strategy was particularly valuable in tasks requiring a nuanced understanding of medical scenarios. CoT prompting guided the model in generating intermediate reasoning steps before arriving at a final answer. Few-Shot Learning complemented the CoT strategy by providing the model with a small number of task-specific examples. These examples demonstrated the reasoning process expected for open-ended tasks, allowing the model to generalize patterns and structures relevant to medical problem-solving. This combination of few-shot learning and CoT was used to encourage models to articulate their reasoning pathways, making evaluating their depth of understanding and logical consistency easier.

Self-consistency, a technique that aggregates multiple independent reasoning paths to identify the most frequent conclusion, was initially considered because of its potential to improve response accuracy. However, this approach was not used because of its high computational cost, mainly when testing large models or conducting extensive evaluations. The additional expense of generating multiple independent answers for each question was disregarded, given the limitations of the study's resources.

## 4.3 Evaluation Metrics

The evaluation of the models in this study was based on three primary metrics: accuracy, energy consumption, and monetary cost.

### 4.3.1 Performance Evaluation Strategies

Accuracy is the primary measure of the models' effectiveness in handling the complex reasoning and factual recall required by the MedQA dataset. For multiple-choice questions, accuracy was measured by comparing the predicted option with the correct one provided in the dataset. In open-answer tasks, semantic similarity scores and LLM-based evaluations were used to determine whether the generated answers aligned with the expected responses.

The evaluation strategies for this study were designed to ensure accurate and consistent assessment of the models' performance across multiple-choice and open-answer formats.

For the multiple-choice questions, the evaluation was based on a simple approach. The code extracted the letter corresponding to the answer predicted by the model and compared it directly with the correct option provided in the dataset. This method was chosen for its simplicity and efficiency, as it eliminated any ambiguity in the evaluation of the answers. By limiting the evaluation to a direct comparison of letters, the process remained consistent and scalable, even when testing multiple models on large datasets. This approach was also well suited to the structured nature of multiple-choice questions, ensuring an objective assessment of the model's ability to select the most correct answer.

The evaluation of the open-ended answers required a more detailed approach to capture the semantic meaning and reasoning behind the model answers. A semantic similarity metric using a BERT model and the Sentence Transformers library was used, with a threshold score of 65% to determine whether the

predicted answer was considered semantically close to the correct answer. This method allowed the evaluation to take into account variations in phrasing and expression, while ensuring that the core meaning of the answer was aligned with the expected answer. The choice of a semantic similarity threshold was informed by experimental tests to balance sensitivity and specificity, ensuring that meaningful responses were correctly identified while minimizing false positives.

To address the shortcomings of semantic similarity metrics, large language models were introduced as evaluators. The evaluator model was prompted to compare the predicted response to the correct answer, both with and without providing the original question. This dual setup allowed the study to measure the impact of contextual information on the evaluation process. The LLM evaluator was instructed to classify responses as correct or incorrect, providing a direct comparison to the ground truth.

Self-consistency was applied to the LLM-based evaluation to improve reliability. Three different LLMs, which were not used in the original tests, were tasked with independently evaluating each response. The final classification of correct or incorrect was determined by majority vote across these models. This approach ensured a robust evaluation process by mitigating biases or inconsistencies from any single evaluator model. The use of self-consistency also improved confidence in the results, as multiple perspectives were considered in the final assessment.

Initially, as mentioned, the evaluation prompts included the question, the predicted answer, and the correct answer, asking the evaluator to classify the response as correct or incorrect. However, it was observed that the evaluator sometimes attempted to generate its own response to the question, introducing bias into the evaluation process.

The question was removed from the evaluator's input to mitigate this issue, leaving only the predicted and correct answers for comparison. This adjustment reduced the evaluator's tendency to infer its own answers and improved the objectivity of the evaluation. This stage established a robust framework for evaluating open-answer responses.

Also, as indicated, our first evaluation framework relied on semantic similarity metrics, using pretrained models such as BERT to compare the predicted answers with the correct ones. However, during the evaluation of the initial selected small dataset, we concluded that this approach had limitations. Many models used synonyms or alternative terms that were semantically equivalent but failed to meet the strict threshold, leading to an underestimation of the models' performance. This realization highlighted the need for a more nuanced evaluation approach, prompting the incorporation of LLM-based evaluation.

Despite improvements, some biases persisted in the evaluation process. This is why, to enhance reliability, self-consistency was introduced, behaving like the aforementioned three LLM output. The final classification of correct or incorrect was determined based on the majority vote among the evaluators. This approach minimized the influence of individual model biases, resulting in a more robust and unbiased evaluation.

As a final step in the evaluation process, an expert doctor reviewed the responses generated by different models and techniques. The expert evaluation provided an additional layer of validation, ensuring that the answers were not only technically accurate but also clinically relevant. This step connected

automated evaluation and real-world applicability, offering valuable insights into the practical use of the models in medical settings.

### 4.3.2 Energy Consumption Quantification

The energy consumption of each model was evaluated to assess its environmental and operational efficiency. Energy consumption was quantified by monitoring the power usage during inference, explicitly measuring the wattage of the hardware for local models and estimating the energy footprint of API requests for service-based models. For local setups, the information of the servers was used while publicly available energy consumption data provided by cloud providers or service APIs helped estimate the energy usage of service-based models. The energy consumption was then aggregated over the evaluation period to calculate the total energy required for each model.

#### 4.3.2.1 Local Models

To quantify the energy consumption of the local models, the NVIDIA System Management Interface (nvidia-smi) tool was employed to monitor and record the energy usage of the GPU during model inference. This tool provides detailed power metrics in watts, allowing for precise tracking of the energy consumed during specific time intervals.

For each local model, the energy consumption process involved the following steps:

1. **Execution of Inference:** Each model was executed with the specified tasks, and the inference process was timed to determine the average duration $t_{\text{avg}}$ of execution for a single question.

2. **Power Consumption Calculation Per Question:** During inference, the `nvidia-smi` tool recorded the average power usage $p_{\text{avg}}$ of the GPU in watts and these measurements were averaged across all questions to provide the typical power draw during active model inference.

3. **Energy Consumption Per Question:** The energy consumed per question was calculated using the formula:

$$E_{\text{question}} = \frac{P_{\text{avg}} \cdot t_{\text{avg}}}{3600 \times 100}$$

   where $E_{\text{question}}$ is the energy consumed per question in kilowatt-hours (kWh), $p_{\text{avg}}$ is the average power usage in watts, and $t_{\text{avg}}$ is the average inference time in seconds.

#### 4.3.2.2 Service-based Models

The energy consumption per query for large language models like GPT-3 and GPT-4 has been a subject of various studies and reports. Although there is no concrete data regarding the energy costs for the most popular service LLMs, it is estimated that a single response using GPT-3, a model with 175 billion parameters, consumes approximately 0.003 kWh of energy. Considering this baseline for the GPT-3

model and its number of parameters, we were able to estimate energy consumption of derivate LLMs based on their number of parameters.

### 4.3.3 Monetary Costs Quantification

Monetary cost was another important metric used to assess the financial feasibility of using each model. For service-based models, the cost was determined by tracking the usage fees associated with API calls. For local models, the monetary cost was estimated based on the cost of GPU instances on AWS, including the hourly rates for the necessary hardware resources.

These values were calculated for each model, offering a standardized basis for comparison across various configurations and setups. By incorporating accuracy, energy consumption, and monetary cost into the evaluation framework, this study provides a comprehensive view of the trade-offs between local and service-based LLMs in the context of medical question answering. This analysis highlights the financial and operational implications of adopting either solution, enabling informed decision-making for healthcare applications.

Table 4.1 below presents the reference values for costs for a instance of AWS in Stockholm GPU configurations similar to those used in the local LLM testing conducted in this study These GPUs provide comparable specifications to those required for AI workloads and natural language processing tasks, ensuring adequate performance for benchmarking. The costs are expressed in US dollars ($) and are categorized into three usage models: On Demand, 1-Year Reservation, and 3-Year Reservation. The table includes cost estimates for different durations (hourly, monthly, yearly, and three years) based on GPU characteristics such as memory and processing power.

| GPUs | Memory (Gb) | €/h | 1 month | 1 year | 3 years |
|---|---|---|---|---|---|
| | | | **On Demand** | | |
| g6.xlarge | 16 | 0,85 $ | 623,13 $ | 7 477,54 $ | 22 432,61 $ |
| g6.2xlarge | 32 | 1,04 $ | 756,94 $ | 9 083,24 $ | 27 249,73 $ |
| g6.4xlarge | 64 | 1,40 $ | 1 024,48 $ | 12 293,78 $ | 36 881,35 $ |
| g6.8xlarge | 128 | 2,14 $ | 1 559,72 $ | 18 716,62 $ | 56 149,85 $ |
| | | | **1 Year Reservation** | | |
| g6.xlarge | 16 | - | 378,58 $ | 4 542,96 $ | 13 628,88 $ |
| g6.2xlarge | 32 | - | 459,92 $ | 5 519,04 $ | 16 557,12 $ |
| g6.4xlarge | 64 | - | 622,50 $ | 7 470,00 $ | 22 410,00 $ |
| g6.8xlarge | 128 | - | 1 015,35 $ | 12 184,20 $ | 36 552,60 $ |
| | | | **3 Year Reservation** | | |
| g6.xlarge | 16 | - | 248,94 $ | 2 987,28 $ | 8 961,84 $ |
| g6.2xlarge | 32 | - | 302,39 $ | 3 628,68 $ | 10 886,04 $ |
| g6.4xlarge | 64 | - | 409,28 $ | 4 911,36 $ | 14 734,08 $ |
| g6.8xlarge | 128 | - | 623,08 $ | 7 476,96 $ | 22 430,88 $ |

Table 4.1: Table de cost for GPU L4 Tensor GPUs in AWS

Table 4.2 below summarizes the costs associated with using different APIs for processing input and generating output tokens. The values are expressed US dollars ($) and represent the cost per 1,000 tokens for both input and output.

| Model | Input (1k tokens) | Output (1k tokens) |
|---|---|---|
| GPT4o | 0,00500 $ | 0,01500 $ |
| GPT4-1106-preview | 0,01000 $ | 0,03000 $ |
| Mistral Small | 0,00200 $ | 0,00600 $ |
| Gemini 1.5 Flash | 0,00035 $ | 0,00105 $ |

Table 4.2: Cost per 1k tokens for input and output across different service-based models.

## 4.4 RAG

### 4.4.1 RAG Implementation

RAG is critical to this research, ensuring the model's outputs are accurate, contextually relevant, and aligned with the domain-specific knowledge required for medical applications.

The prompt for this implementation was carefully designed to guide the LLM's behaviour, ensuring strict adherence to the provided context and enhancing the interpretability of outputs, resulting in better performance. The prompt structure is as follows:

```
INSTRUCTIONS: As a medical assistant, only use information from the context.
If the context doesn't answer the question, propose two questions by only changing
medical terms based on the context. Answer in the same language as the question.
If you don't know the language, answer in English.

begin of english context: {context}.
end of context. {completion}
```

This prompt serves multiple purposes:

1. **Role Specification:** The phrase *"As a medical assistant"* establishes the model's role, aligning its behaviour with the expectations of a medical professional.

2. **Context Restriction:** The explicit directive to *"only use information from the context"* ensures that the model's responses remain grounded in the retrieved data, minimizing the risk of hallucinations.

3. **Fallback Mechanism:** If the context is insufficient, the model is instructed to propose two alternative questions by altering medical terminology, establishing the system's capability to reframe queries in ways that can lead to better-informed answers in subsequent interactions, for instance:

   - Original Question: *What are the side effects of Drug X?*
   - Context: Insufficient information on Drug X.
   - Generated Questions:
     (a) *What are the side effects of medications in the same class as Drug X?*
     (b) *What are the common side effects associated with similar treatments?*

4. **Multilingual Support:** Requires that the answer is in the same language as the question, broadening accessibility and applicability in diverse linguistic settings. Furthermore, the fallback to English ensures the output is understandable, even in edge cases where unsupported languages are encountered.

The context `{context}` embedded in the prompt is then dynamically populated using a retrieval system, which indexes and searches relevant medical datasets or documents, ensuring that only the most pertinent information is provided to the model after going through the following steps:

1. **Dataset Preparation:** Medical datasets used in the retrieval system are pre-processed and indexed for efficient querying. These datasets include authoritative resources such as clinical guidelines, medical textbooks, and peer-reviewed articles.

2. **Query Processing:** Incoming user queries are mapped to search terms using natural language processing techniques, extracting relevant context from the indexed datasets.

3. **Context Injection:** Retrieved snippets are inserted into the `{context}` placeholder of the prompt, forming the basis for the model's response generation.

This retrieval step ensures the generative model is equipped with domain-specific information, bridging the gap between static knowledge and real-time applicability.

### 4.4.2 RAG Parameter Variations

Our RAG approach uses two parameters, the number of documents retrieved from the underlying knowledge (*top-k* search) and number of retrieved documents that are re-ranked for relevance and passed to the generation phase (*top-x* rerank), to control the retrieval and ranking processes that underpin its ability to generate accurate and contextually relevant responses. In this study, we explored a range of values for these parameters, varying *top-k* search from 10 to 100 and *top-x* rerank from 3 to 40, to better understand their impact on model performance and to optimize their configuration for the MedQA dataset.

A higher value of documents retrieved from the knowledge base during the retrieval phase increases the breadth of information retrieved, providing the model with a broader context for generating answers. However, retrieving too many documents can introduce irrelevant or noisy data, potentially overwhelming the model and reducing its ability to focus on the most relevant content. By testing a range of values, we aimed to identify the optimal trade-off between retrieval breadth and relevance, ensuring that the models received sufficient context without compromising quality.

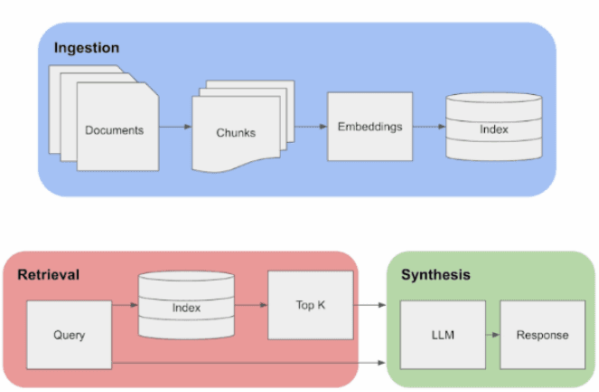


Figure 4.2: RAG Pipeline

Defining how many of the retrieved documents are re-ranked for relevance and passed to the generation phases is crucial for filtering out less pertinent information and ensuring that the most relevant documents are prioritized in the generation process. Smaller values focus on a highly selective subset of documents, which can improve precision but risks excluding useful contextual information. Larger values provide a more comprehensive context for generation but can dilute relevance, particularly if the retrieval phase has already introduced noise. The choice to vary *top-x* rerank allowed us to assess how ranking depth affects the accuracy and reasoning capabilities of the models.

The decision to explore these parameters over a wide range of values reflects the need to account for the variability in question complexity and document relevance inherent to the MedQA dataset. Medical questions often require multi-hop reasoning and integration of diverse information sources, making it important to optimize both retrieval and ranking processes. By systematically varying k top search and k top rerank, we could evaluate their influence on the models' ability to retrieve and utilize the most relevant information effectively.

This comprehensive exploration also enabled insights into the computational trade-offs associated with these parameters. Larger values for *top-k* search and *top-x* rerank generally increase processing time and resource usage, which is particularly important when benchmarking local and service-based models. Understanding these trade-offs is crucial for determining configurations that balance performance and efficiency, especially in real-world applications where computational constraints often come into play.

# Chapter 5

# Tests and Results

This chapter presents an overview of all experiments conducted on the two selected datasets, from data and model selection, to scenario outlining and the testing results, including performance and cost analysis.

## 5.1 Testing setup

### 5.1.1 Dataset Selection

The MedQA dataset (presented in Subsection 3.3.4.1) is one of the most recent datasets created for the evaluation of medical question-answer systems. Proposed by [139], it features a multiple choice format. This format is suitable for quantitative evaluation as it enables a direct comparison of performance between different language models, facilitating the benchmark analysis in this study.

The MedQA dataset is sourced from the United States Medical Licensing Examination, making it representative of real-world medical problem-solving. The balance of the dataset across medical specialties ensures a comprehensive evaluation of medical knowledge, and its uniform distribution of answer choices minimizes bias. However, its complexity presents challenges, as multi-hop reasoning and diverse specialties can be particularly difficult for models with limited context windows or reasoning capabilities.

An initial smaller dataset comprising 11 entries (from now on named "Smaller Dataset") was used for the experiments to allow a careful and individualized evaluation of how the models responded to the questions. The initial step involved identifying questions related to Parkinson's disease from the MedQA dataset. A set of keywords, including "tremor," "motor fluctuations," "parkinson disease/diagnosis," and "parkinsonian disorders," was used to filter the dataset for relevant questions. This keyword-driven search resulted in the mentioned subset of 11 questions, which provided a manageable and focused dataset for starting the code development process. Each question was carefully examined to ensure its relevance to the topic and its suitability for testing the capabilities of the selected models. This approach

enabled iterative improvements to the process, including adjustments to the prompting strategies and evaluation methods.

Once these strategies were refined and defined, testing proceeded with a larger dataset (from now on named "Larger Dataset"): a 1400-line dataset for multiple-choice questions and a 500-line dataset for open-ended questions. These larger datasets were employed to assess the models' performance on more extensive and diverse tasks, ensuring a robust evaluation framework.

### 5.1.2 Model Selection

Selecting the appropriate local and service-based models was a critical step in implementing the benchmarking process. The selected service-based models were GPT-4o, GPT-1106-Preview, Gemini-1.5-Flash, and Mistral-Small. These models were selected for their position as state-of-the-art solutions and their robust performance across a range of general-purpose and domain-specific tasks. GPT-4o offered a cost-efficient alternative with relevant reasoning abilities, while GPT-1106-Preview provided insights into experimental capabilities of upcoming GPT variants. Gemini-1.5-Flash combined retrieval-augmented generation techniques with advanced language modeling, making it particularly relevant for medical QA. Mistral-Small was chosen for its lightweight nature and cost-efficiency, offering a counterpoint to the larger and more resource-intensive models.

The selected local models selected were Mistral-Nemo, Mistral, LLaMA 3.2-3B, LLaMA 3.1, MedLLAMA 2, Meditron-7B Gemma2-9B and Phi4. Their diverse capabilities, parameter scales, and specific task optimisations guided these models' selection. These models represent a broad spectrum of state-of-the-art local language models, encompassing general-purpose models such as LLaMA 3.2-3B and LLaMA 3.1, as well as domain-specialized models such as MedLLAMA 2 and Meditron-7B, which is fine-tuned for medical applications. The inclusion of Mistral and Mistral-Nemo offers lightweight, high-performance options ideal for cost-efficient and low-latency tasks, while Gemma2-9B and Phi4 provide larger-scale alternatives capable of handling complex reasoning and multi-turn interactions. This diverse selection ensures a comprehensive benchmarking study by evaluating models of varying sizes and specializations, allowing for a detailed comparison with service-based models in terms of performance, cost-efficiency, and domain-specific accuracy.

By selecting models from these distinct categories, we ensure a representative sample of the LLM landscape, encompassing high-performance, proprietary systems and more customizable, resource-efficient alternatives. This diversity enables an analysis of how various hosting paradigms and architectural strategies impact performance, scalability, and usability in real-world applications. Additionally, it allows for a robust exploration of trade-offs between cost, accessibility, and functionality, providing valuable insights for stakeholders with varying priorities and constraints.

## 5.2 Scenarios Description

To evaluate the performance of local and service-based LLMs under varying conditions, eight different testing scenarios were designed. These scenarios are summarized in Table 5.1 and explore combinations of dataset sizes, question types, evaluation methods, and model deployment types. The integration of RAG across all scenarios ensured consistent retrieval of relevant contextual information, allowing for a robust assessment of each model's ability to utilize external knowledge effectively.

Table 5.1: Testing Scenarios and Parameters

| **Scenario** | **Smaller Dataset** | **Larger Dataset** | **Multiple Choice** | **Open-Answer** | **Expert Review** | **Local Models** | **Service-Based Model** |
|---|---|---|---|---|---|---|---|
| S1 | X | | X | | | X | |
| S2 | X | | X | | | | X |
| S3 | X | | | X | X | X | |
| S4 | X | | | X | X | | X |
| S5 | | X | X | | | X | |
| S6 | | X | X | | | | X |
| S7 | | X | | X | | X | |
| S8 | | X | | X | | | X |

The scenarios were designed to address specific aspects of model performance:

- **Smaller Dataset Scenarios** focus on initial testing and improvement of evaluation strategies with a limited number of questions (the aforementioned 11).

- **Bigger Dataset Scenarios** test the scalability and robustness of the models across a broader set of questions.

- **Question Types** include both multiple choice formats for precise evaluation and open-answer formats to test the reasoning capabilities of the models.

- **Model Types** compare local and service-based deployments to assess their relative strengths and limitations in different contexts.

For the multiple-choice tasks, the prompt used was:

```
Answer the following question by selecting one of the options provided.

 Question:
 {question}

 Options:
 {options_str}

```

```
 Instructions:
 - Review the options carefully.
 - Provide your answer in the following format: 'Answer: [Option Letter]'.
 - Do not include any additional text or explanation."

 Answer:
```

For the open-answer tasks, the prompt used was:

```
Example 1:
Question: A 30-year-old woman reports fatigue, weight gain, and cold intolerance.
    Physical examination reveals dry skin and bradycardia. TSH levels are elevated, and
    free T4 is low.

Question: What is the most likely diagnosis?

Step-by-Step Answer:
Symptoms like fatigue, weight gain, and cold intolerance suggest a metabolic slowdown,
    typical of hypothyroidism.
Dry skin and bradycardia further support this hypothesis.
Lab results show elevated TSH (the body trying to stimulate the thyroid) and low free T4
    , confirming primary hypothyroidism.

Example 2: ...
Example 3: ...

Now, answer the following question step-by-step:
{question}
```

The tests for the service-based models were conducted using the available APIs for each model. Specifically, the Gemini 1.5 Flash model was accessed via the Google Cloud AI API, GPT-4o and GPT-4 Preview were evaluated using the OpenAI API, and Mistral Small was tested through the MistralAI. On the other hand, the tests for the local models were executed on a L4 Tensor Core GPUs using the Lightning AI platform.

**S1: Smaller Dataset with QA and Local Models**

This scenario utilized the smaller dataset to evaluate the performance of local models in answering multiple choice questions. The limited dataset allowed for careful observation of model behavior in simpler, controlled tasks, providing a foundation for refining evaluation strategies. RAG was employed to enhance the retrieval of domain-specific knowledge, enabling a targeted assessment of the models' performance in simpler, structured tasks.

**S2: Smaller Dataset with QA and Service-Based Models**

Similar to Scenario 1, this scenario employed a smaller dataset but focused on service-based models. The objective was to compare their performance to local models in handling domain-specific multiple-choice questions under similar conditions. The use of RAG allowed for a direct comparison of retrieval and reasoning capabilities between local and service-based models under controlled conditions.

**S3: Smaller Dataset with Open-Answer and Local Models**

In this scenario, local models were tested on a smaller dataset consisting of open-ended questions. RAG was utilized to provide relevant context for generating free-text responses, enabling an evaluation of the models' generative capabilities in unstructured tasks.

**S4: Smaller Dataset with Open-Answer and Service-Based Models**

This scenario mirrored Scenario 3 but tested service-based models on the same smaller dataset with open-ended questions. By employing RAG, the models were equipped to generate more informed and coherent responses, facilitating a comparison with local models in generative tasks.

**S5: Larger Dataset with QA and Local Models**

Expanding the dataset to include a larger number of questions, this scenario focused on local models and MCQA formats. Expanding to the larger dataset, this scenario tested local models on multiple-choice tasks. RAG was critical in retrieving a broader range of contextual information, helping to assess the scalability and robustness of the models.

**S6: Larger Dataset with QA and Service-Based Models**

Using the same larger dataset as Scenario 5, this scenario evaluated service-based models in multiple-choice tasks. The integration of RAG ensured consistency in retrieval and reasoning, enabling a fair comparison with local models.

**S7: Larger Dataset with Open-Answer and Local Models**

Local models were tested with the larger dataset of open-ended questions in this scenario. RAG played a pivotal role in providing extensive contextual information, allowing the models to handle more diverse and complex generative tasks effectively.

**S8: Larger Dataset with Open-Answer and Service-Based Models**

The final scenario explored the use of service-based models for open-answer questions with the larger dataset. By utilizing RAG, the models were able to integrate external knowledge into their responses, highlighting their adaptability in unstructured tasks requiring significant reasoning and generative capabilities.

## 5.3 Results and Analysis

This section provides an in-depth analysis of the experimental results obtained from testing various LLM configurations across different scenarios. The goal of this analysis is to assess the performance of service-based and local LLMs in tasks with varying complexities and configurations (RAG parameter variations mentioned in Subsection 4.4.2).

### 5.3.1 S1: Smaller Dataset with QA and Local Models

Figure 5.1 provides a comparison of model accuracy across various configurations of top-k search and rerank values. Phi-4 emerges as the top-performing model, achieving the highest accuracy (up to 100%) in configurations with top-k search=6 and top-k rerank=5, showcasing its exceptional capability to retrieve and synthesize relevant information. LLaMA 3.1 and LLaMA 3.2-3B also perform remarkably well, achieving accuracies above 81% in optimal configurations, demonstrating their strong architectural optimization for retrieval-augmented tasks. Similarly, Mistral NeMo, with the same 128k context window as the LLaMA models, delivers comparable performance, achieving up to 72.73%, although it slightly lags behind in key configurations.

Gemma 2-9B shows weaker performance, particularly for configurations with higher rerank values, where its accuracy drops. This suggests that the model struggles to effectively utilize larger amounts of retrieved information, likely due to limitations in its context window lenght. Mistral, with a smaller context window, performs consistently lower, rarely exceeding 54.55%. MedLLaMA 2, across all configurations, performs poorly.

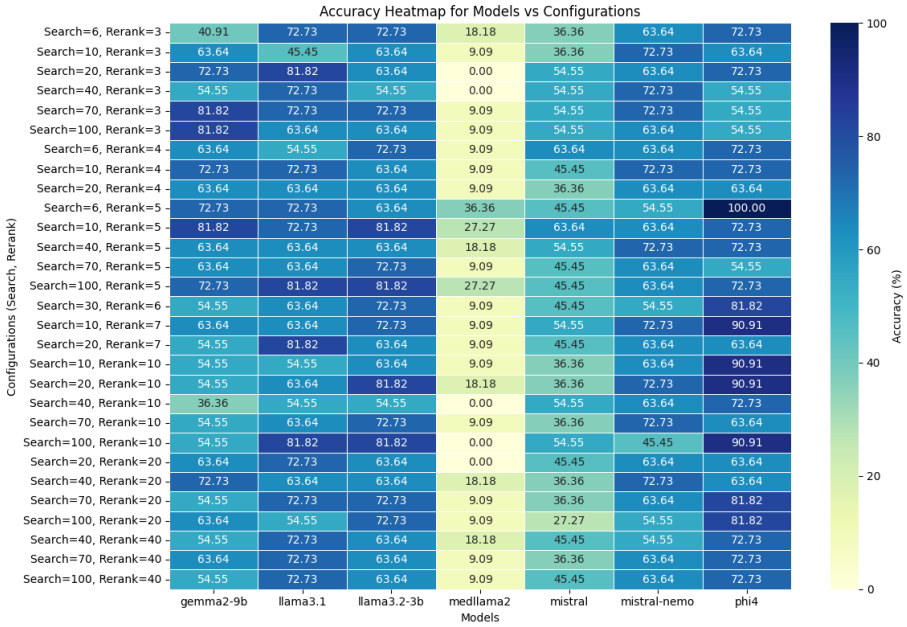


Figure 5.1: Results of S1: Smaller Dataset with QA and Local Models

### 5.3.2 S2: Smaller Dataset with QA and Service-Based Models

Figure 5.2 The heatmap presents a comparison of model accuracy across different configurations of top-k search and rerank values for various models. GPT-4-1106-preview consistently achieves the highest accuracy, reaching 90.91% in configurations with higher values of top-k rerank, showing its superior optimization for retrieval-augmented tasks. In contrast, GPT-4o demonstrates lower accuracy, with results that closely align with those of Gemini-1.5 Flash. Both GPT-4o and Gemini-1.5 Flash achieve a maximum accuracy of 81.82% in specific configurations, such as top-k search=10 and top-k rerank=7. On the other hand, Mistral Small, with its 4k context window, consistently delivers the lowest accuracy, even in configurations with higher rerank values, reflecting its limitations in leveraging larger retrieval outputs effectively.

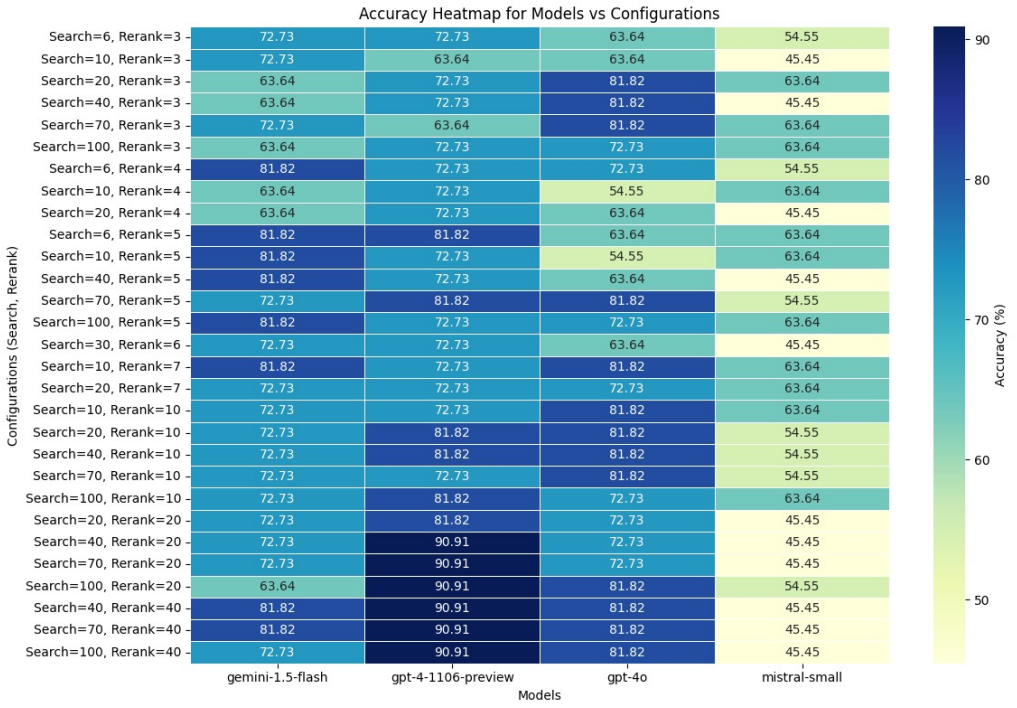


Figure 5.2: Results of S2: Smaller Dataset with QA and Service-Based Models

### 5.3.3 S3: Smaller Dataset with Open-Answer and Local Models

Figure 5.3 provides a comparative analysis of model accuracy for open-answer tasks across different top-k search and rerank configurations using local models. Phi-4 demonstrates the highest accuracy among the models, achieving a peak of 54.55% in lower configurations such as top-k search=6 and rerank=4 and also in higher ones, such as top-k search=20 and rerank=20, showcasing its capability to handle open-answer tasks effectively. In contrast, Mistral NeMo achieves moderate performance, with its accuracy consistently hovering around 40% in the majority of configurations, indicating some limitations in processing open-ended responses. LLaMA 3.1 and LLaMA 3.2-3B deliver comparable results, with a maximum accuracy of 54.54% in configurations with top-k search=100. Gemma 2-9B, however, struggles with higher top-k rerank values, showing inconsistent performance and rarely surpassing 27.27%, suggesting difficulties in effectively leveraging larger retrieval outputs. Mistral (7B) and MedLLaMA 2 show the lowest performance across configurations, with accuracy frequently below 18.18%, reflecting their challenges in addressing the demands of open-answer tasks. LLaMA 3.1 and LLaMA 3.2-3B deliver comparable results, with a maximum accuracy of 54.54% in configurations with top-k search=100. Gemma 2-9B struggles with higher rerank values, showing inconsistent performance and rarely surpassing 27.27%. Mistral delivers consistently low accuracy, typically below 18.18%, reflecting its limited capacity for open-answer tasks. MedLLaMA 2, however, performs worst, with results consistently at 0% across all configurations, emphasizing its inability to address open-ended

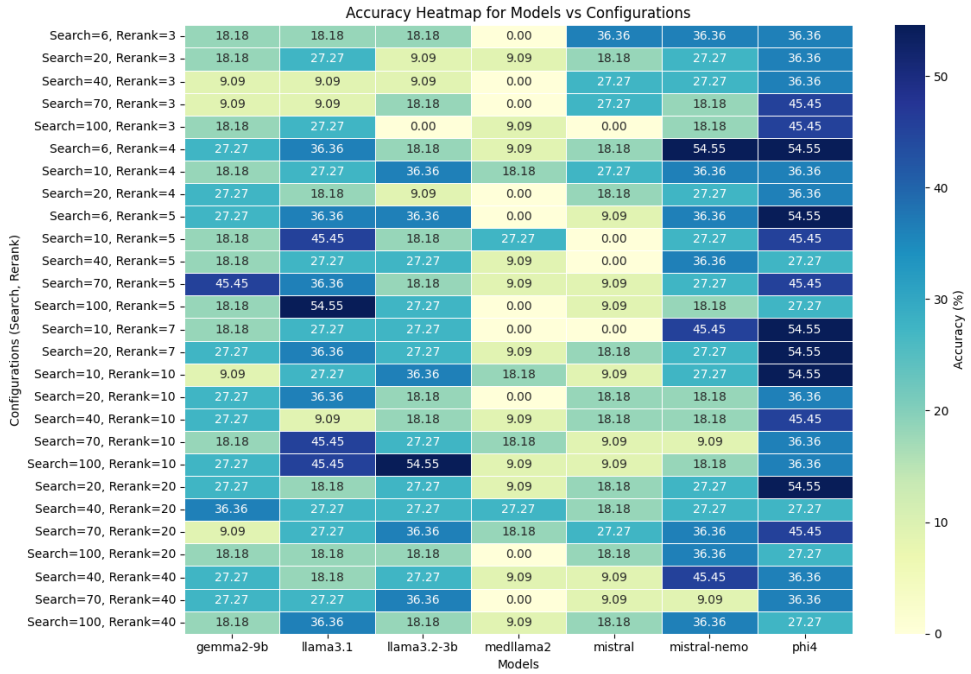


Figure 5.3: Results of S3: Smaller Dataset with Open-Answer and Local Models

question-answering tasks effectively.

### 5.3.4 S4: Smaller Dataset with Open-Answer and Service-Based Models

Figure 5.4 presents the accuracy results of service-based models tested on a smaller dataset with open-ended questions, evaluated across varying configurations of Top Search and Top Rerank parameters.

Phi-4 consistently demonstrates the best performance, achieving the highest accuracy of 57.21% in configurations such as search=40 and rerank=5, showing its robust optimization for retrieval-augmented tasks. Gemma 2-9B emerges as the second-best model, reaching a peak accuracy of 45.21% in configurations such as top-k search=10 and rerank=10, demonstrating its ability to effectively handle retrieval outputs and maintain competitive performance. LLaMA 3.2-3B achieves moderate accuracy with high rerank values, peaking at 44.29% in configurations such as search=40, rerank=10 but it performs significantly worse in configurations with lower rerank values, with accuracy dropping to 21.29% in setups like search=40, rerank=3. Mistral NeMo, shows moderate performance with a maximum accuracy of 38.29% in configurations like search=10, rerank=10, falling behind Gemma 2-9B. Finally, Mistral exhibits the weakest performance, rarely exceeding 30.93%, reflecting its limitations in processing and synthesizing larger retrieval outputs effectively.

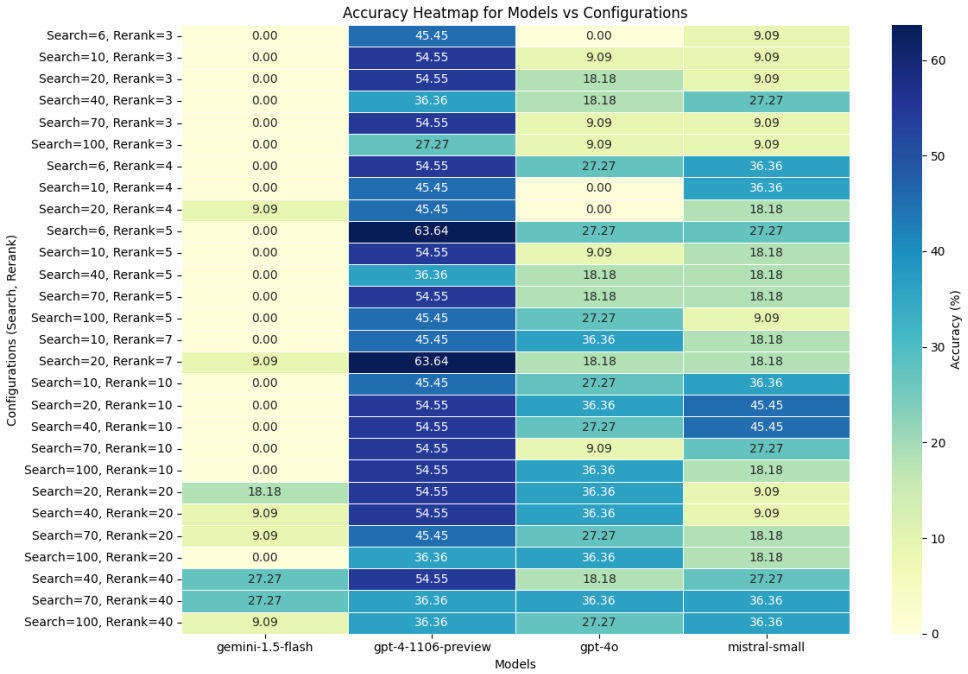


Figure 5.4: Results of S4: Smaller Dataset with Open-Answer and Service-Based Models

### 5.3.5 S5: Larger Dataset with QA and Local Models

The heatmap compares the accuracy of various models across different configurations of top-k search and rerank values for QA tasks on the larger dataset. Phi-4 consistently demonstrates the best performance, achieving the highest accuracy of 57.21% in configurations such as search=40 and rerank=5, showing its robust optimization for retrieval-augmented tasks. Gemma 2-9B emerges as the second-best model, reaching a peak accuracy of 45.21% in configurations such as top-k search=10 and rerank=10, demonstrating its ability to effectively handle retrieval outputs and maintain competitive performance. LLaMA 3.2-3B delivers consistent results, achieving a maximum accuracy of 44.29% in configurations such as top-k search=40 and top-k rerank=10. Mistral NeMo, shows moderate performance with a maximum accuracy of 38.29% in configurations like search=10, rerank=10, falling behind Gemma 2-9B. Finally, Mistral exhibits the weakest performance, rarely exceeding 30.93%, reflecting its limitations in processing and synthesizing larger retrieval outputs effectively.


Figure 5.5: Results of S5: Larger Dataset with QA and Local Models

### 5.3.6 S6: Larger Dataset with QA and Service-Based Models

The heatmap in Figure 5.6 illustrates the performance of local models when tested on the larger dataset with multiple choice questions.

GPT-4-1106-preview demonstrates the highest performance, achieving a peak accuracy of 66.14% in the configuration top-k search=40 and rerank=10. GPT-4o, while achieving a solid maximum accuracy of 57.69% in the configuration top-k search=40 and rerank=3, exhibits less consistent results overall, with noticeably lower performance in other configurations, reflecting challenges in maintaining reliability across setups. In contrast, Gemini-1.5 Flash delivers more consistent results across configurations. While its peak accuracy of 46.36% (in top-k search=40 and rerank=10) is lower than the best results achieved by GPT-4 models, its performance remains relatively stable across all configurations, generally outperforming GPT-4o in scenarios with lower accuracy. Mistral Small, with a maximum accuracy of 42.57%, continues to underperform, reflecting its architectural and scalability limitations.

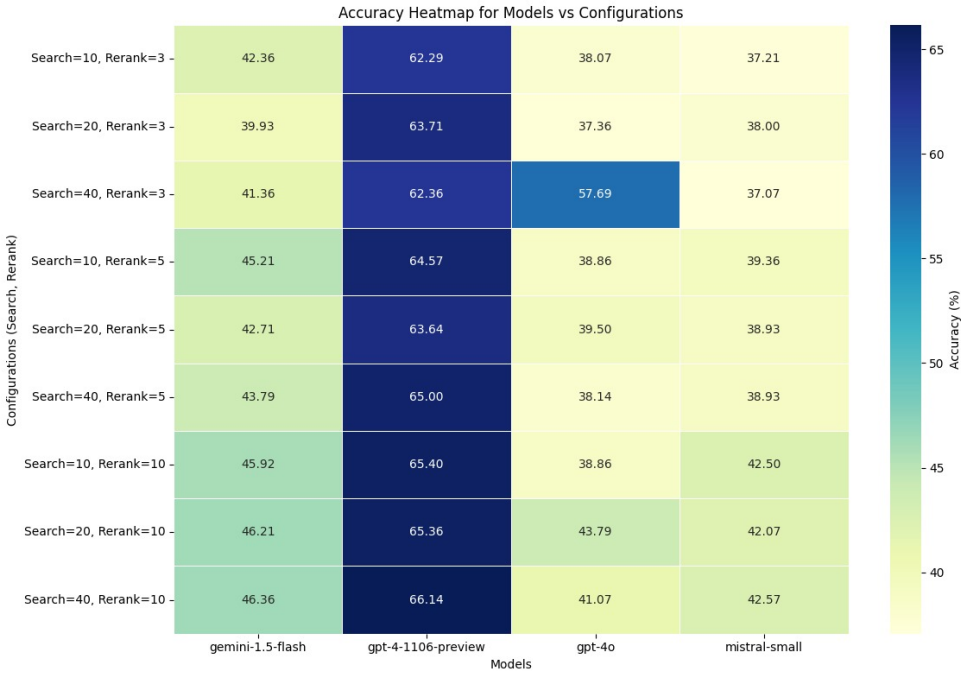


Figure 5.6: Results of S6: Larger Dataset with QA and Service-Based Models

### 5.3.7 S7: Larger Dataset with Open-Answer and Local Models

In Figure 5.7, the results of the bigger dataset with open-answer questions using local models are presented. Among the tested models, Phi4 achieves the highest accuracy at 33.0%, outperforming the other models by a significant margin. The next best-performing models, Gemma2-9B Mistral-NeMo, achieve an accuracy of 14.6% and 14.0%, respectively, which is less than half of Phi4's performance. Following this, Mistral demonstrates an accuracy of 9.6%, indicating moderate performance in this context. The lowest accuracy is observed for LLaMA 3.2-3B, with only 4.4%, highlighting a substantial gap in performance compared to the other models. These results indicate a considerable variation in performance among the local models for the open-answer task, with Phi4 being the most accurate, while LLaMA 3.2 (3B) lags significantly behind.

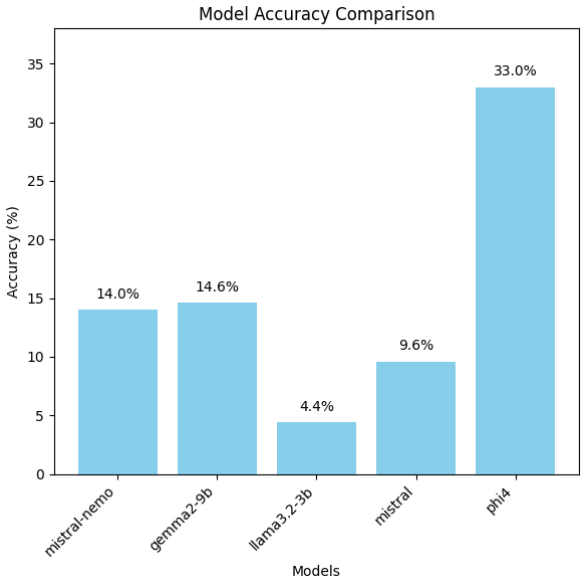


Figure 5.7: Results of S7: Larger Dataset with Open-Answer and Local Models

### 5.3.8 S8: Larger Dataset with Open-Answer and Service-Based Models

In Figure 5.8, the results of the larger dataset with open-answer questions using service-based models are presented. Among the tested models, GPT-4-1106 Preview achieves the highest accuracy, with a value of 32.%, significantly outperforming the other service-based models. The second-best performing model, Mistral Small 11.0%, achieves a markedly lower accuracy of 11.0%, illustrating a substantial gap in performance compared to GPT-4-1106 preview. Following this, GPT-4o demonstrates an accuracy of 7.8%. Lastly, Gemini-1.5-Flash records the lowest accuracy of all, with only 0.2%, indicating minimal effectiveness in handling this open-answer task. These results reveal a significant variation in the capabilities of service-based models in addressing open-ended questions within a larger dataset. GPT-4-1106-Preview stands out as the most competent, while the performance of the other models, particularly Gemini-1.5-Flash, is substantially lower, emphasizing a wide performance disparity across the service-based models tested.

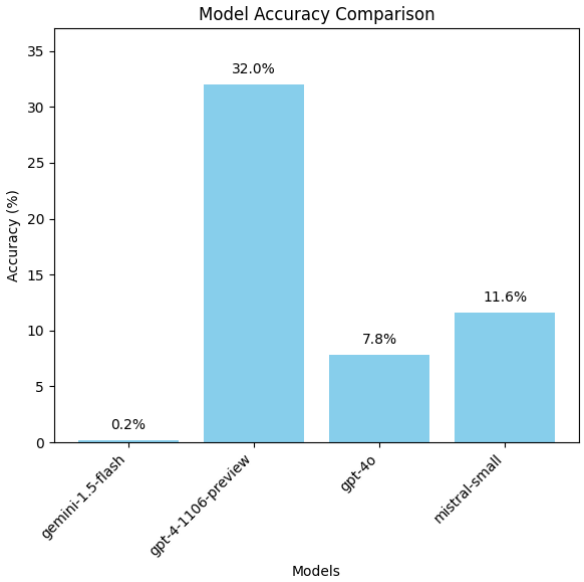


Figure 5.8: Results of S8: Larger Dataset with Open-Answer and Service-Based Models

### 5.3.9 Expert Review

To ensure the reliability and efficiency of the evaluation method employed for open-ended questions in the smaller dataset (focused on Parkinson-related topics), a subset of these questions was randomly selected and carefully assessed by a neurologist. The evaluation method under review utilized a large language model to automatically determine the correctness of the answers provided by other LLMs. The neurologist independently reviewed each of these questions and their corresponding evaluations, classifying them as correctly or incorrectly assessed by the automated method.

The analysis revealed that **72%** of the questions were correctly evaluated by the LLM-based evaluation method. This indicates a reasonably high level of alignment between the automated evaluation and the expert neurologist's judgment. However, the results also highlight that approximately **28%** of the evaluations were incorrect, suggesting areas where the evaluation method may need refinement, particularly in nuanced medical scenarios. These findings underscore the importance of incorporating expert validation in such studies, as it provides a critical benchmark for understanding the strengths and limitations of automated evaluation techniques in complex, domain-specific tasks.

### 5.3.10 Energy Consumption and Costs

#### 5.3.10.1 Local Models

Table 5.2, shows the results of running local models in a A100-SMX4-40GB GPU, based on the leaderboard of [142].

Table 5.2: Energy Consumption of Local Models with A100-SXM4-40GB GPU

| **Model** | **Parameters (Billions)** | **Energy per Response (Joules)** | **Energy per Request (kWh)** |
|---|---|---|---|
| Phi4 | 14 | 86.08 | 0.0000239 |
| Lamma3.2 - 3b | 3 | 2.26 | 0.0000006 |
| Mistral 7B | 7 | 43.76 | 0.0000125 |
| Llama 3.1 8B | 8 | 51.12 | 0.0000122 |
| Mistral Nemo | 12 | 66.71 | 0.0000185 |
| Gemma 2 9B | 9 | 68.24 | 0.0000190 |

Following the formula defined in Subsection 4.3.2.1, the following energy costs per request were calculated for the models and GPU used in this study.

Table 5.3: Energy Consumption of Local Models with L4 Tensor GPU

| **Model** | **Parameters (Billions)** | **Energy per Response (Joules)** | **Energy per Request (kWh)** |
|---|---|---|---|
| Llama 3.2 - 3B | 3 | 59.34 | 0.0000165 |
| Mistral 7B | 7 | 62.94 | 0.0000175 |
| Gemma 2 9B | 9 | 63.65 | 0.0000177 |
| Mistral Nemo | 12 | 59.82 | 0.0000166 |
| Phi 4 | 14 | 64.76 | 0.0000180 |

#### 5.3.10.2 Service-Based Models

As discussed in Chapter 4, these LLMs do not provide precise or reliable data regarding the energy costs associated with their services. However, it is known that there is some correlation between the billions of parameters in the model and the energy consumed in Joules per response.

To estimate the energy costs of service-based LLMs for comparison purposes, the results obtained from local LLMs on each GPU were analyzed. Using this data, the linear regression model in Figure 5.9 was defined to establish a relationship between the number of billions of parameters and the energy cost for the A100-SXM4-40GB GPU.

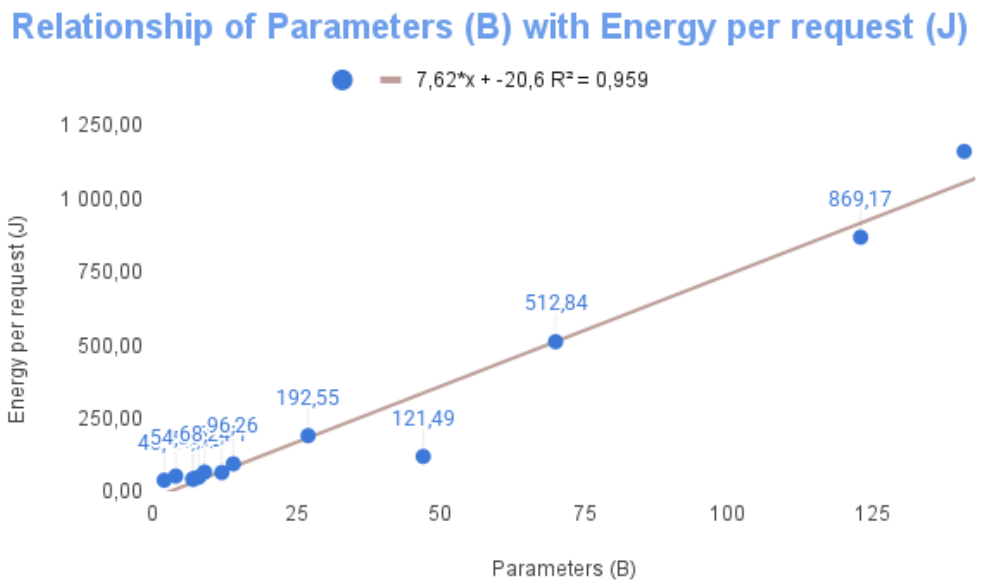


Figure 5.9: Linear Regression Model for energy cost calculations in A100-SXM4-40GB GPU

The computed regression function yielded the calculations in Table 5.4.

GPT-3 calculations are included here as it is the only model for which an energy estimation could be found. According to [143], GPT-3 is estimated to consume 0.0003 kWh per request, which is closely aligned with the results from this regression, calculating the energy cost at 0.00027 kWh per request. This reinforces the validity of this estimation technique for extrapolating energy consumption values for other models, enabling a fair comparison between local and service-based models.

Table 5.4: Energy Consumption of Service Models with A100-SXM4-40GB GPU

| **Model** | **Parameters (Billions)** | **Energy per Response (Joules)** | **Energy per Request (kWh)** |
|---|---|---|---|
| GPT-3 | 128 | 954.76 | 0.0002652 |
| GPT-4o | 1,370 | 10,418.8 | 0.0028941 |
| Mistral-Small | 22 | 147.04 | 0.0000408 |
| GPT-4-1106-preview | 1,750 | 13,314.4 | 0.0036984 |
| Gemini-1.5-Flash | - | - | - |

Using the same approach, the regression function presented in Figure 5.10 was obtained for the L4 Tensor GPU (used to test the local models in this study).

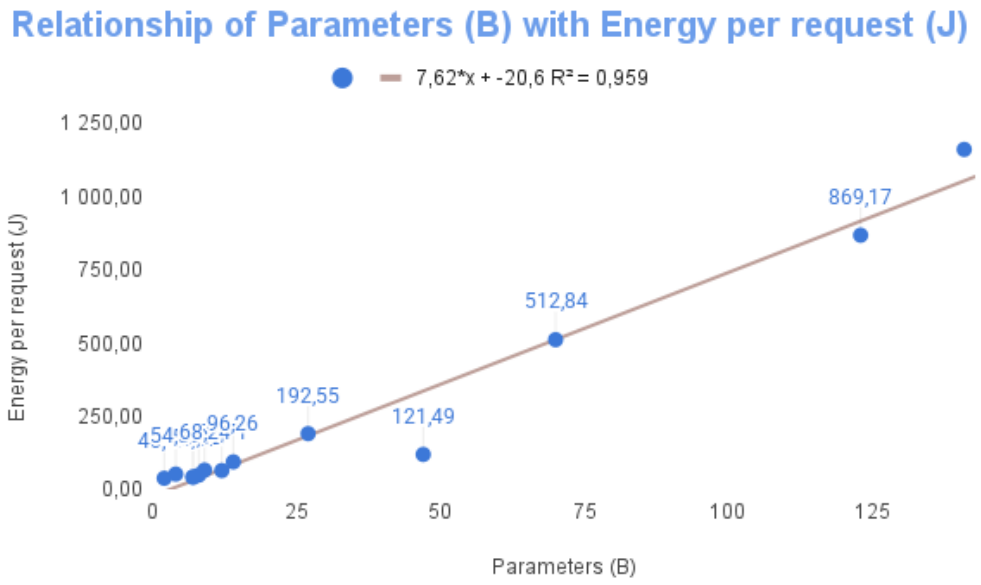


Figure 5.10: Linear Regression Model for energy cost calculations in L4 Tensor GPU

The computed regression function produced the calculations in Table 5.5.

Table 5.5: Energy Consumption of Service Models with L4 Tensor GPU

| **Model** | **Parameters (Billions)** | **Energy per Response (Joules)** | **Energy per Request (kWh)** |
|---|---|---|---|
| GPT-3 | 128 | 96.364 | 0.0000268 |
| GPT-4o | 1,370 | 454.06 | 0.0001261 |
| Mistral-Small | 22 | 65.836 | 0.0000183 |
| GPT-4-1106-preview | 1,750 | 563.50 | 0.0001565 |
| Gemini-1.5-Flash | - | - | - |

Gemini-1.5-Flash had no available information regarding the amount of parameters, so it was not possible to estimate the costs for this use case.

### 5.3.11 Monetary Costs

The monetary costs of local and service-based LLMs differ significantly due to their different deployment and pricing models. Understanding these differences is critical to determining the most cost-effective solution for specific use cases, especially when considering scalability. The values presented for APi costs were calculated based on an input sentence and its corresponding output, considering these as average values. The input consists of 20 words, while the output contains 100 words. These word counts were converted to tokens using an average estimation of tokens per word, resulting in approximately 26.67 tokens for the input and 133.33 tokens for the output.

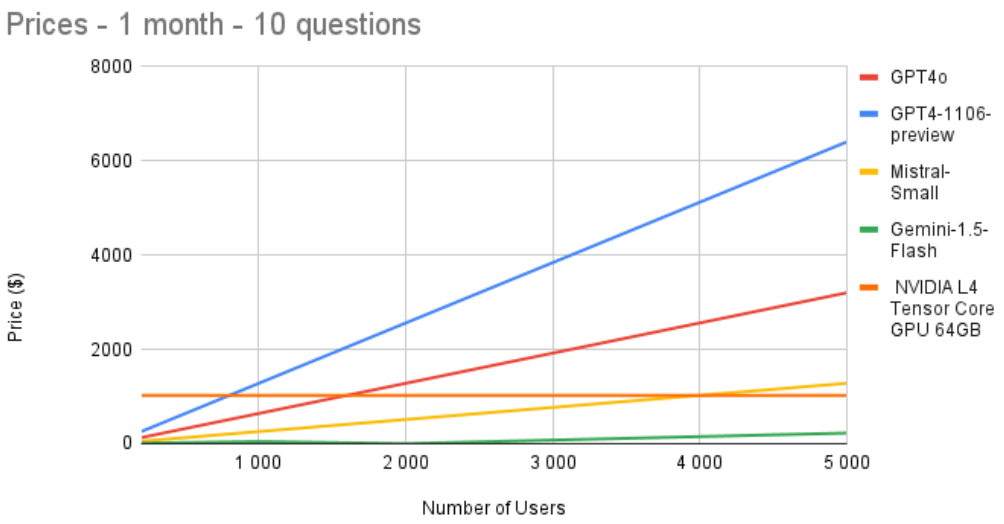


Figure 5.11: Linear Chart of Cost Evolution in relation to the number of users after one month

Figure 5.11 illustrates the cost variation over one month as the number of users increases, assuming an average of 10 questions per user daily. This visualization highlights how costs scale differently between local and service-based models. Service-based models, such as GPT-4-1106-preview, exhibit higher cost growth rates compared to local solutions like Mistral-Small and GPU L4 Tensor Core.

Over a longer period, the financial disparity between the models becomes more pronounced. Figure 5.12 shows the cost variation over one year, under the same assumptions of 10 daily questions per user and Figure 5.13 shows the cost variation over three years for 5 daily questions per user.

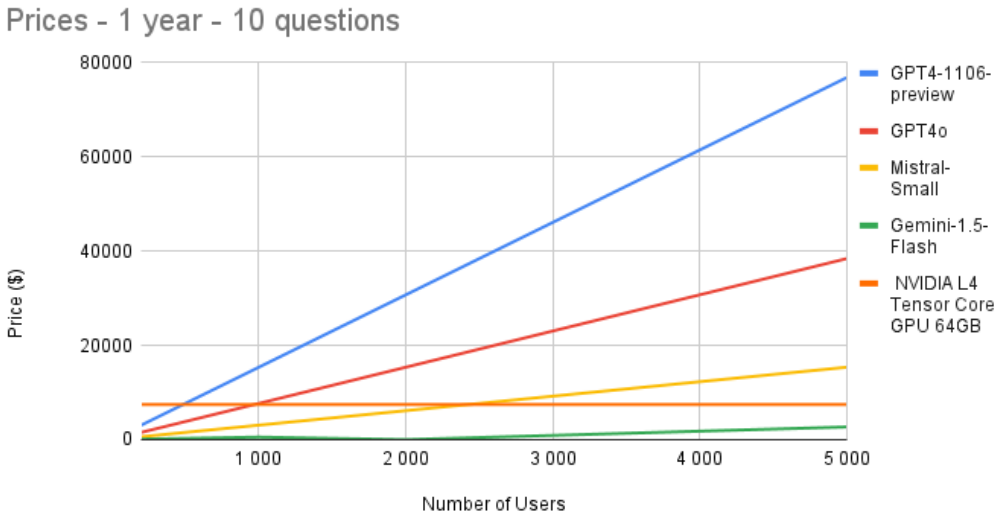


Figure 5.12: Linear Chart of Cost Evolution in relation to the number of users after one year

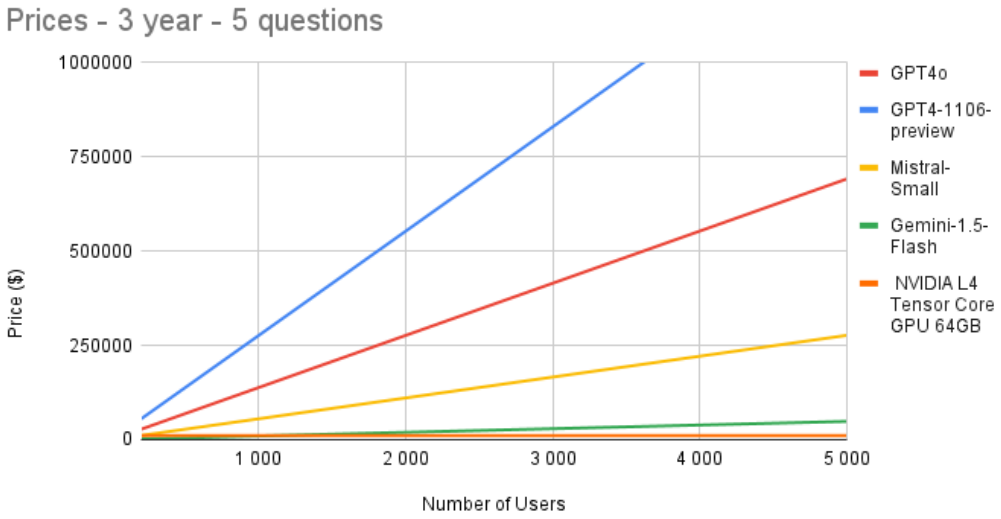


Figure 5.13: Linear Chart of Cost Evolution in relation to the number of users after three years

# Chapter 6

# Discussion

This chapter analyzes the findings from evaluating local and service-based LLMs in conjunction with RAG. It highlights the trade-offs between performance, cost, privacy, and scalability, particularly in healthcare applications. Key challenges, such as prompt sensitivity and computational demands, are examined alongside the benefits of RAG in improving domain-specific accuracy.

## 6.1 Initial dataset

The results presented in the last chapter from scenarios S1 to S4 illustrate the performance of various local and service-based models across different configurations of RAG tasks, with the initial smaller dataset. This discussion analyzes the results, focusing on the characteristics of each model, the impact of their configurations, and the implications of context window size and parameter count on their accuracy.

### 6.1.1 Local Models

The LLaMA-based models, **LLaMA 3.2 (3B)** and **LLaMA 3.1 (8B)** are designed with a context length of 128k tokens, which allows them to handle RAG tasks effectively. In the QA task, both models show relatively consistent performance across different configurations of search and rerank values, with accuracy ranging from 45.45% to 81.82%. Neither model demonstrates substantial sensitivity to increasing search values, provided that reranking is appropriately configured. Across all configurations, the accuracy of LLaMA 3.2 closely matches that of LLaMA 3.1, suggesting that the additional parameters in the larger model do not translate into significant improvements in QA performance with RAG. In the open-ended task, both models display significantly lower accuracy compared to QA, which is expected given the increased complexity of synthesizing responses for open-ended questions. The best performance for LLaMA 3.2 occurs in settings with *top-k search* of 100 and *top-x rerank* of 10 and for LLaMA 3.1 in settings with *top-k search* of 100 and *top-x rerank* of 5, with an accuracy of 54.55%.

However, in smaller configurations, both models struggle. Given these results, it is evident that the performance of LLaMA 3.2 (3B) and LLaMA 3.1 (8B) is comparable in both QA and open-ended

tasks. Both models share the same architecture and context length, but the larger parameter size of LLaMA 3.1 does not yield significant performance improvements. As a result, LLaMA 3.2 (3B) was deemed more resource efficient and was selected for further evaluation on the larger dataset. Meanwhile, LLaMA 3.1 (8B) was excluded from subsequent tests to prioritize exploration of the more parameter-efficient model. This decision ensures a balanced focus on performance and computational efficiency in subsequent analyses.

The analysis of the smaller dataset revealed significant limitations in the performance of **MedLLaMA 2**, particularly in tasks that involve multiple options and open questions. For multiple-choice questions, the model consistently failed to follow the explicit instruction in the prompt to select a single option objectively: to choose a specific letter corresponding to the correct answer. Instead, it provided lengthy and descriptive responses without explicitly selecting an option. Since the automated evaluation of multiple choice questions is based on identifying the correct letter in the response of the model, this inability to comply with the prompt led to very low accuracy scores for MedLLaMA 2 in this task. For open-ended questions, MedLLaMA 2 demonstrated a frequent lack of focus on the specific question being asked. For instance, in questions that required identifying potential factors leading to a diagnosis, the model often responded with recommendations on how the patient should manage or resolve the presented condition. This deviation from the intended focus of the question further undermined the model's utility in these scenarios. Additionally, MedLLaMA 2 has a limited context window of 4096 tokens, restricting its ability to effectively utilize larger contexts that might be required to answer more complex questions or synthesize information across multiple retrieved documents. Given these limitations in both multiple-choice and open-ended tasks, coupled with its restricted context window, we decided not to proceed with MedLLaMA 2 for tests on the larger dataset. This decision was made to ensure the reliability and comparability of results across models and configurations better suited for these tasks.

**Mistral 7B**, with a context window of 4096 tokens and an extended processing capacity of 8192 tokens, achieves moderate results, particularly in configurations with low rerank values, where its compact architecture allows efficient processing of smaller retrieval outputs. However, as the retrieval complexity increases, its performance plateaus because of its relatively limited context window and inability to integrate large amounts of retrieved information effectively. However, with the same context size and number of parameters as MedLLaMA 2 it consistently achieved superior results.

**Mistral NeMo 12B** supports a 128k context window, similar to LLaMA 3.1 and LLaMA 3.2, but this model performs slightly worse than LLaMA 3.1 and LLaMA 3.2 in QA tasks. It achieves its highest accuracy of 72.73%, comparable to the LLaMA models, in configurations with values of top k search of 70 and top k rerank of 10. These results suggest that the size of the context window plays a more significant role in achieving strong performance in RAG tasks than the number of model parameters. Despite substantial differences in parameter sizes: LLaMA 3.1 with 8 billion, LLaMA 3.2 with 3 billion, and Mistral NeMo with 12 billion, the models deliver comparable accuracy across both QA and open-ended tasks. All three models share a 128k-token context window, which allows them to process and integrate large retrieval outputs effectively. This demonstrates that the ability to handle long contexts is a key fac-

tor for performance in retrieval-augmented setups, while increasing the number of parameters does not necessarily yield significant gains in these tasks. The results highlight that architectural optimizations, such as context window size, may provide more impactful improvements than simply scaling up model size.

With 9 billion parameters and a context window of 8k tokens, **Gemma2 (9B)**, for QA tasks, performs better when fewer documents are reranked such as values of 3 or 5 documents. These results suggest that the model is more effective when provided with a smaller, more focused subset of retrievals. For open-ended tasks, the model performance remains relatively consistent across all configurations, regardless of the search or rerank values.

### 6.1.2 Service Based Models

The results obtained for service-based models highlight the trade-offs between context window size, parameter count, and overall accuracy. The selected models, GPT-4.0, GPT-4 1106 Preview, Mistral Small, and Gemini 1.5 Flash, demonstrate distinct behaviors influenced by their architectural characteristics and design.

**Gemini-1.5 Flash**, with a window context of 1 million tokens, consistently delivers moderate performance across all configurations, with accuracy values remaining within the 54.55% range across configurations with higher rerank values for QA taks. The lack of significant variation across configurations suggests limited adaptability to changes in search and rerank parameters. The performance of Gemini-1.5 Flash significantly drops for open-answer tasks, with accuracy values as low as 0% in configurations with low search or rerank parameters. Even in the best-case scenario, *top-k search* = 20 and *top-x* rerank = 40, the accuracy reaches only 27.27%, highlighting a limited capability to generate open-ended responses. The model's reliance on retrieval alone without effective generative reasoning may explain these results.

**GPT-4-1106 Preview** achieves top-tier accuracy (90.91%) in several configurations, particularly for higher rerank values. The model's 128k token context window and extensive parameter count (about 175 trillion) ensure superior contextual comprehension, which translates into robust QA performance. The model maintains strong performance in open-answer tasks, achieving 63.64% precision under configurations with high rerank values, *top-k* search = 20 and *top-x* rerank = 20, but this represents a noticeable decrease compared to QA tasks, likely due to the increased complexity of generating coherent and contextually rich open-ended answers. Its large parameter count and extended context window contribute to its overall reliability in all configurations.

**GPT-4o** exhibits moderate performance, peaking at 81.82% accuracy for configurations with higher rerank values and moderate search parameters. Its performance aligns closely with GPT-4-1106 Preview but does not match the latter's peak accuracy. This is likely due to GPT-4o's reduced parameter count or differing optimization strategies for the RAG architecture. The model's 8k token context window could serve as a limiting factor when handling more complex retrieval tasks requiring extended contextual understanding. Similar to QA tasks, GPT-4o demonstrates moderate accuracy in open-answer tasks,

with peak accuracy at 36.36% for configurations with higher rerank values, *top-k* search = 40 and *top-x* Search = 40. The model struggles more in open-ended contexts compared to GPT-4-1106 Preview due to its smaller parameter size (about 135 trillions).

**Mistral Small** demonstrates significantly inferior performance compared to other models in QA tasks, despite its more compact architecture with a smaller context length of 32k. Its limited context length may restrict its ability to fully leverage larger retrieval outputs in structured tasks, leading to lower accuracy. Furthermore, its smaller parameter size, in contrast to the much larger-scale models from OpenAI, further contributes to its challenges in precision-focused tasks like QA. However, in open-ended tasks, Mistral Small delivers results comparable to GPT-4o and far superior to Gemini 1.5 Flash, indicating potential shortcomings in the latter. The stronger performance of Mistral Small in open-ended tasks can be justified by its capacity to generate coherent and contextually relevant responses, despite handling smaller retrievals. Unlike Gemini 1.5 Flash, which often fails to focus on the core question or adhere to the prompt's intent, Mistral Small appears more capable of synthesizing and integrating retrieved information within its smaller context, demonstrating an architectural advantage in handling reasoning-intensive tasks, even with its more constrained resources.

#### 6.1.2.1 Conclusions

The results obtained from the smaller dataset were used for the selection of the configurations for the subsequent tests conducted on the larger dataset, by analyzing the performance of various top-k search and top-k rerank configurations. This selection optimizes for both performance and computational efficiency, ensuring robust evaluation of local and service-based models across diverse tasks and maximizing the capabilities of RAG while addressing the strengths and limitations of each model type.

- **Search Values:**
  - A search depth of **10** ensures quick and efficient retrieval of relevant candidates. This setting proved to be sufficient for QA tasks where precision is crucial, and excessive retrieval depth does not yield significant accuracy gains.
  - A search depth of **20** strikes a balance between retrieval breadth and computational cost. It showed strong performance across both QA and open-answer tasks, enabling deeper exploration of potential answers without incurring excessive processing times.
  - A search depth of **40** optimizes for complex, open-answer scenarios where a broader retrieval scope significantly improves generative reasoning by providing more comprehensive candidate inputs.
- **Rerank Values:**
  - A rerank value of **3** a lightweight yet effective ranking mechanism, suitable for tasks where speed is prioritized without sacrificing too much accuracy.

- Reranking **5** candidates balances ranking precision with computational efficiency, showing improvements in QA tasks by refining the top candidates for final selection.
- A rerank value of **10** is recommended for open-answer tasks, where refining a larger set of candidates is critical for achieving higher generative accuracy.

Based on the results of the expert review, which indicated that 72% of the evaluations from the smaller dataset were accurate, self-consistency was implemented in the LLM evaluators to enhance the reliability of the evaluation process for open-ended questions in the larger dataset. The models Gemini 1.5 Flash, Mistral Small, and GPT-4 Turbo were utilized as LLM evaluators, generating multiple evaluations for each answer. By applying self-consistency to the evaluators, this adjustment addressed potential inaccuracies and ensured a more reliable assessment of the models' performance in handling complex medical questions.

## 6.2 Final dataset

The results presented in the last section from scenarios S5 to S8 illustrate the performance of various local and service-based models across different configurations of RAG tasks, with the final larger dataset. In this section of the discussion we analyze the results, focusing on the characteristics of each model, the impact of their configurations, and the implications of context window size and parameter count on their accuracy.

### 6.2.1 Service-Based Models

The **GPT-4-1106-Preview** model achieves the highest accuracy across all configurations, consistently outperforming the other models. With accuracies peaking at 66.14% for *top-k* search of 40 and *top-x* rerank of 10, GPT-4-1106-Preview demonstrates its ability to leverage extensive retrieval outputs and rerank configurations effectively. This strong performance is indicative of its fine-tuning for retrieval-augmented tasks and robust architecture, which enables it to process and integrate large amounts of contextual information. The model's high accuracy across configurations also reflects its capacity for precise and coherent answer generation, making it the most reliable model for QA tasks in this evaluation.

In the open-ended diagnostic tasks, **Mistral-Small** achieved 11% accuracy, outperforming GPT-4o, which obtained 7.8%, and Gemini-1.5-flash, which achieved only 0.2%. Despite its smaller architecture, Mistral-Small's ability to marginally surpass these larger models highlights its relative strength in handling specific aspects of open-ended tasks, possibly due to its optimization for concise reasoning and response generation. However, it remains significantly behind GPT-4-1106-preview, which achieved 33% accuracy, reflecting the limitations of Mistral-Small in tasks requiring complex contextual synthesis and advanced reasoning. In the multiple-choice tasks, Mistral-Small's performance was comparable

to GPT-4o and only slightly lower than Gemini-1.5-flash. This result underscores Mistral-Small's relative competence in structured retrieval and reasoning for straightforward multiple-choice scenarios, achieving results that align with larger models. However, it still lacks the contextual depth and precision integration demonstrated by more advanced models like GPT-4-1106-preview, particularly in configurations with increased search and reranking parameters.

The **Gemini 1.5 Flash** model's contrasting performance in open-ended versus multiple-choice tasks can be attributed to the inherent differences in the cognitive demands of these formats and the model's architectural strengths and limitations. In multiple-choice tests, the model benefits from the structured nature of the task, where predefined options provide contextual anchors that help narrow its focus and improve decision-making accuracy, aligning with the [144] study's findings that Gemini 1.5 Flash excels at retrieving and processing relevant information when the output space is well-defined and constrained. Conversely, open-ended questions require the model to independently generate coherent and contextually appropriate responses without explicit guidance from predefined options. This increases reliance on the model's ability to synthesize retrieved information effectively and manage ambiguities in the input. The study highlights that Gemini 1.5 Flash often struggles with these aspects, particularly due to its tendency to produce incomplete answers or unnecessarily activate safety filters. This suggests that while the model is well-suited for structured tasks like multiple-choice evaluations, it faces significant challenges in handling the unstructured nature of open-ended questions, irrespective of RAG configuration size tests in this study.

The unexpected results of **GPT-4o** in these tests, with a performance in QA tasks comparable to Mistral-Small and slightly below Gemini 1.5 Flash, and underperforming Mistral-Small in open-ended tasks, can be attributed to several factors. Studies suggest that GPT-4o, despite being a large and capable model, faces challenges in RAG scenarios due to its architectural trade-offs. The study in [145] refers that large models with extensive internal knowledge, like GPT-4o, often rely less on external retrieval, which may hinder their ability to fully utilize RAG techniques when context integration is required. Furthermore, the study [144] highlights that larger models may struggle with synthesizing retrieved information effectively in certain domains, which could explain its subpar performance in tasks requiring nuanced reasoning and contextual synthesis compared to smaller, more domain-optimized models like Mistral-Small and Gemini 1.5 Flash. Additionally, the inherent design of GPT-4o may prioritize broader generalization over specialized performance, leading to limitations in open-ended tasks such as those in medical diagnostics. These findings collectively underscore GPT-4o's unexpected results, as its architectural and optimization choices can be less suited to the specific demands of your benchmark.

### 6.2.2 Local Models

The results for the local LLMs reveal significant variations in performance between the models tested, highlighting differences in their capabilities to handle complex question-answering (QA) tasks. The **Phi4** model achieved the highest accuracy, with 33% in open-answer tasks and strong performance in structured QA tasks, outperforming all other models. This demonstrates its ability to effectively retrieve

and integrate relevant context as well as synthesize accurate responses, particularly in larger datasets where complex reasoning is required. This superior performance is likely attributed to its larger parameter size (14 billion) and advanced training strategies, which enhance its capacity to process intricate questions and contextual dependencies.

**Gemma2-9b** achieved a moderate accuracy of 14.6%, reflecting its competency in RAG techniques for QA tasks. However, its performance is significantly lower than that of Phi4, suggesting limitations in its ability to manage the complexity and breadth of the larger dataset. This could be due to a less sophisticated architecture or limited fine-tuning for domain-specific knowledge integration.

**Mistral-Nemo**, demonstrated moderate performance on QA tasks, achieving a maximum accuracy of approximately 38.3% in QA tasks. While its extended context capacity allows it to handle large retrievals effectively, its performance remains below of models such as Phi-4. Despite its potential for processing long contexts and 12 billion parameters, Mistral NeMo's results suggest that it does not fully leverage this capability to outperform other local models like LLaMA 3.2-3B in terms of accuracy, highlighting room for improvement in its optimization for RAG tasks.

**Llama3.2-3b**, with only 4.4% accuracy in open-ended questions, performs poorly compared to most local models, reflecting limitations in its ability to handle larger datasets effectively. Its small architecture and reduced capacity contribute to its difficulty in retrieving and synthesizing relevant information, especially when faced with more complex questions.

**Mistral**, with 4096 context window, with 9.6% accuracy in open-answer tasks, demonstrates relatively modest performance, which is consistent with expectations for a smaller-scale model. In QA tasks, achieves a maximum accuracy of 30.8%, and LLaMA 3.2-3B consistently outperforms it across higher configurations, despite having only 3 billion parameters compared to Mistral's 7 billion. This discrepancy underscores the critical impact of context window size on performance in retrieval-augmented tasks, as the ability to process and integrate larger amounts of retrieved information appears to outweigh the advantage of additional parameters. The results highlight that Mistral's limited context window is a significant constraint.

Overall, the results suggest that larger and more advanced models, such as Phi4, are better suited to tasks involving extensive datasets, as they can manage the computational and contextual challenges these datasets present. In contrast, smaller models, while performing adequately in structured QA scenarios, lack the depth and sophistication needed for open-answer tasks in complex and expansive datasets. These findings emphasize the importance of model size, training, and optimization in achieving higher performance in large-scale QA applications.

These results demonstrate that RAG can indeed help bridge the performance gap between smaller models and those with much larger scales. Notably, Gemma2-9b model achieved results comparable to the significantly larger Gemini-1.5 Flash in QA tasks, highlighting the potential of RAG techniques to enhance the capabilities of smaller models. Moreover, the exceptional performance of Phi4, a local model with 14 billion parameters, is particularly noteworthy. Phi4 outperformed all other models in open-answer tests, including GPT-4-1106-preview, the top-performing service-based model in previous

tests. This achievement highlights the potential of well-optimized local models to compete with and even surpass service-based models when paired with robust retrieval mechanisms and domain-specific fine-tuning.

## 6.3 Energy Consumption

This section provides a comparative analysis of the energy measurements collected for each model (both service-based and local) when generating responses. The data include the number of parameters (in billions), the estimated energy per response (in joules), and an approximate conversion to kWh per request. Tables 5.2, 5.3 , 5.4 and 5.3 present the measurements.

At a general level, models with larger parameter counts, such as GPT-4-1106-preview (1.75 trillion parameters) and GPT4o (1.37 trillion parameters), consume more energy per inference than smaller models such as the local LLMs tested in this study. For instance, the distinction is clearly observable in GPT-4-1106-preview's energy per response of roughly 1313 versus Mistral-Small's 147J. Although there is a general correlation between parameter size and energy, the relationship is not strictly linear—some mid-range local models (e.g., 12B vs. 14B) have similar or even slightly lower/higher measured energy, proving that other factors (such as hardware optimizations, model architecture, or inference batch sizes) can also affect power draw.

The results show that large commercial APIs, particularly GPT-4 variants, have substancially higher energy footprints per request, likely due to the massive scale of underlying infrastructure and the complexities of distributed inference. In contrast, local models such as phi4, LLaMa3.2 (3B), or mistral exhibit much smaller energy usage, with around 60J per response on the most efficient GPU model. This difference is due to their smaller parameter footprints, but these values could be even further apart due to the possibility that local inference configurations (especially if optimized or run on specialized hardware) could be more efficient than large-scale distributed cloud systems.

GPT-4-1106-preview and GPT4o, with parameter counts in the trillions, can yield energy consumption reaching hundreds or even thousands of joules per response and although these models offer top-tier accuracy, the energy cost per request is notably higher. Phi4 measured at 64.76J, remaining significantly more economical in energy terms than trillion-parameter service models—roughly one to two orders of magnitude less. The smaller local architectures have results substantially lower than their trillion-scale counterparts.

In summary, the energy measurements underscore a major trade-off between large models, which can consume over an order of magnitude more energy per response, and smaller local models, which are generally far more energy-efficient but sacrifice some performance. However, the Phi4 achieved very similar performance to the GPT-4-1106-preview models, and has significantly lower energy expenditure, demonstrating that it is possible to maintain accuracy while reducing energy consumption. Although the number of parameters loosely correlates with higher power usage, specific implementation details and inference methods can significantly modulate actual consumption.

## 6.4 Monetary Costs

This section analyzes the monetary costs associated with deploying LLMs, comparing GPU infrastructures suitable for local inferences against service-based or API-driven solutions. The presented data encompass both AWS pricing (for different GPU instances under on-demand or reserved contracts) and the projected monthly and yearly costs of invoking selected service-based models via API. The primary objective is to identify the scenarios in which one approach may be financially more advantageous than the other and to highlight the factors thamonetary cost differences between service-basedificant differences in monetary costs between service-based LLMs and local LLMs when scaled to varying user numbers and timeframes. The inclusion of the NVIDIA L4 Tensor Core GPU (64GB) in the evaluation provides a important context, as it represents the minimum hardware requirement necessary to run the largest local model, Phi4, due to its substantial memory needs.

Service-based models like GPT-4o and GPT-4-1106-preview exhibit a linear cost increase proportional to the number of users. However, GPT-4-1106-preview's pricing escalates significantly faster than GPT-4o, making it the most expensive option across all timeframes, especially at scale. This high cost reflects its advanced architecture and high compute demands, optimized for premium performance. Over three years with 5,000 users, the cumulative costs of GPT-4-1106-preview approach $1,000,000, demonstrating the financial costs of relying on such models for large-scale deployment.

Local models, when hosted on AWS using the NVIDIA L4 GPU, offer a scalable cost structure that remains competitive for larger user bases. The GPU costs in AWS hosting remain substantially lower than the costs of GPT-4-1106-preview for long-term, high-scale deployments. For example, Phi4, the largest local model in the comparison, requires the memory capacity of the NVIDIA L4 GPU 64GB but still incurs lower costs than GPT-4-1106-preview when scaled to thousands of users over multiple years. However, reliance on cloud-hosted infrastructure for local models introduces recurring operational costs, narrowing the cost advantage over time compared to local hardware ownership.

**High-Volume, Long-Term Scenarios**

For scenarios involving large user bases or continuous high throughput, such as 5,000 users making 10 queries daily, the costs associated with service-based models like GPT-4-1106-preview can become prohibitive, reaching over $1,000,000 per year. In such cases, hosting a local model like Phi4 on an AWS NVIDIA L4 instance becomes a more viable option. Although the initial or monthly costs for GPU infrastructure may seem high, they remain predictable over time and can be amortized with consistent use. These setups also allow for better control over operational costs as the user base scales.

**Low-Volume, Short-Term Scenarios**

When the query volume is low or sporadic, API-based solutions such as GPT-4o or GPT-4-1106-preview are more cost-effective due to their pay-per-call pricing model. These services eliminate the

overhead of idle GPU costs. For example, if the user base consists of fewer than 1,000 users or has irregular activity, the total yearly cost of service-based models remains manageable compared to the fixed costs of hosting local models.

**Mid-Volume, Moderate Growth Scenarios**

For moderate user bases (2,000 to 3,000 users) with steady but not overwhelming growth, local models like Phi4 or Mistral-Small remain a strong choice. Although these models require infrastructure investments, such as AWS-hosted NVIDIA GPUs or on-premises hardware, they offer predictable scaling costs and greater control over operations. Local hosting avoids the exponential cost increases of API-based models as the user base grows while ensuring data remains under control. Additionally, with moderate demand, local models can handle workloads without incurring idle costs, making them a cost-efficient solution in these scenarios.

**Healthcare and Compliance-Critical Scenarios**

In domains such as healthcare, where patient confidentiality and regulatory compliance are paramount, hosting local models like Phi4 becomes the most suitable option. This setup ensures that data remain under direct control, mitigating risks associated with transferring sensitive data to external API-based services. Additionally, local hosting allows better handling of peak usage scenarios, such as hospital systems encountering spikes during emergencies, without incurring unexpected API costs or risking compliance violations.

## 6.5 Medical LLMs vs General LLMs

One of the goals of this dissertation is to compare the performance of medical LLMs and general-purpose LLMs, particularly in the context of RAG. Studies such as [146] have systematically explored the adaptation of general LLMs to the medical domain, highlighting the methodologies and challenges involved in this specialization. Additionally, [147] has also evaluated the efficacy of RAG in enhancing the performance of LLMs on medical question-answering tasks, providing insights into how both medical-trained and general-purpose models leverage external knowledge sources.

In this study, the application of RAG with medical-trained local LLMs proved unfeasible. Specifically, the Meditron 7B model consistently failed to generate responses, resulting in null outputs. This behavior may be attributed to limitations in its training data or architecture, which make its ability to process and retrieve relevant information not very effective. Similarly, the MedLLaMA 2 model did not adhere to the provided prompts, leading to responses that were misaligned with the intended instructions and rendering the evaluation process ineffective. These issues underscore the challenges inherent in deploying medical-trained LLMs within a RAG framework, particularly concerning their responsiveness and alignment with user prompts.

In conclusion, while medical-trained LLMs hold promise for specialized applications, their integration into RAG systems presents significant challenges. Future work should focus on improving the training methodologies and architectural designs of these models to ensure their effective deployment in RAG-based applications within the medical domain.

## 6.6 Trade-Offs between Local and Service Models

This section provides a discussion of how model performance, financial cost, and energy consumption interact when comparing locally hosted LLMs versus service-based solutions, highlighting the primary trade-offs, particularly relevant in highly regulated sectors such as healthcare.

### 6.6.1 Performance vs. Financial Costs

Although models like Gemini-1.5 Flash offer notably low operational costs compared to other LLMs, their performance in open-answer tasks, achieving only 0.2% accuracy, makes them unsuitable for medical applications. In scenarios requiring accurate, contextually rich, and domain-specific responses, such as medical question-answering, the extreme limitations of Gemini-1.5 Flash undermine its utility, regardless of its affordability. For healthcare solutions, where precision and reliability are critical, the trade-off between cost and performance is untenable in this case.

In contrast, higher-performing models like GPT-4-1106-preview deliver exceptional accuracy in open-answer tasks (32%) but at a significant financial cost, particularly in high-volume scenarios. Its pay-per-request structure, while convenient for low-usage scenarios, results in costs that can exceed $1,000,000 annually for large-scale deployments. This makes it a premium option best suited for applications where budget constraints are secondary to performance needs. Phi4 achieved the highest accuracy (33%) in open-answer tasks, demonstrating the potential to rival top-tier service-based APIs while offering predictable costs when hosted on infrastructure like AWS's NVIDIA L4 Tensor Core GPUs. Although the operational costs for running Phi4 are higher than Gemini-1.5 Flash, its superior accuracy justifies the expense, particularly in medical use cases where response quality directly impacts outcomes. Mid-range local models such as Gemma2-9B strike a balance between cost and performance, offering reasonable accuracy and the ability to run on less resource-intensive GPU instances. This makes them a viable option for applications with moderate performance requirements and continuous usage patterns. However, their accuracy still trails larger commercial APIs, which may limit their suitability for more demanding medical tasks. Lastly, smaller API-hosted models such as Mistral-Small provide cost savings for applications with lower performance thresholds, but although affordable, these models may not meet the accuracy requirements of domains such as healthcare, where errors in understanding or generating responses have significant consequences.

The choice between models must carefully weigh performance against costs, especially for medical applications where accuracy and reliability are non-negotiable. Phi4 emerges as a strong contender due to its superior performance and predictable cost structure. For organizations prioritizing affordability,

smaller models like Mistral-Small may suffice for less critical tasks, but for high-stakes scenarios, the investment in higher-performing models such as GPT-4 variants or optimized local LLMs is justified.

### 6.6.2 Performance vs. Energy Cost

Balancing performance and energy consumption is a critical consideration when selecting lLLMs for medical applications. Service-based models such as GPT-4-1106-preview deliver exceptional accuracy (32% in open-answer tasks) but come with a high energy cost of 13,314.4 Joules per response. Similarly, GPT-4o, while slightly less accurate and more efficient at 10,418.8 Joules per response, still exhibits significant energy demands. These values highlight the trade-off between the state-of-the-art performance offered by large-scale service models and the substantial environmental and operational costs associated with their energy usage.

In comparison, local models such as Phi4 achieve similar or even superior performance (33% accuracy) while being much more energy efficient. With an energy consumption of only 86.08 Joules per response, Phi4 consumes approximately 155 times less energy than GPT-4-1106-preview per request. This positions Phi4 as a highly attractive option for high-performance, energy-conscious applications. Other local models like Gemma2-9B and Mistral Nemo strike a balance, consuming 68.24 Joules and 66.71 Joules per response, respectively, while providing adequate performance for moderate workloads. However, smaller local models like Llama3.2-3b, with energy consumption as low as 2.26 Joules per response, excel in energy efficiency but are unsuitable for high-stakes medical tasks due to their significantly lower accuracy and reliability.

**High Performance vs. High Energy**

GPT-4-1106-preview dominates in accuracy but imposes a heavy energy burden, making it optimal for critical tasks where cost and energy efficiency are secondary considerations.

**Balanced Performance and Energy**

Models like Phi4 deliver competitive accuracy while maintaining significantly lower energy consumption, making them a compelling choice for organizations that prioritize sustainability alongside performance.

**Energy Efficiency vs. Accuracy**

Smaller local models, such as Llama3.2-3b and Mistral 7B, offer unparalleled energy efficiency but sacrifice the performance needed for complex medical tasks, limiting their applicability to less critical scenarios.

The trade-off between performance and energy consumption is evident across the spectrum of LLMs. Service-based models provide optimal accuracy but come at the cost of high energy usage, making them suitable only for scenarios where performance outweighs all other factors. Phi4 offer a near-optimal balance, excelling in both accuracy and energy efficiency, making it ideal for sustainable, high-performance

deployments. For lower-stakes or resource-constrained environments, smaller local models such as Llama3.2-3b may suffice, provided the reduced accuracy aligns with the application requirements.

### 6.6.3 Conclusions

Local LLMs are preferable for cases of large-scale and high-intensity usage, with continuous high-volume workloads justify the upfront investment, as cost per request and energy usage might stabilize at manageable levels. Moreover, healthcare settings dealing with sensitive patient data must opt for local solutions to minimize legal complexities, even if it entails higher operational overhead. Service LLMs are suitable for sporadic or low-volume usage, and the pay-per-request model can be more cost-effective, sparing the expense of idle GPU cycles. Ultimately, the optimal choice between local and service-based LLMs depends on the intersection of performance needs, budget availability, energy considerations, and compliance obligations. In the healthcare context, privacy concerns tilt the balance toward on-premises solutions, but the feasibility also depends on whether usage levels and the required accuracy threshold justify resource investments.

# Chapter 7

# Conclusions and Future Work

This chapter concludes this dissertation by reviewing and addressing the research questions originally outlined. Then it summarizes the main contributions of this research, highlighting its importance and potential impact on the field. Finally, the chapter explores the prospects for future directions, presenting opportunities for research and advancement based on the foundations established in this work.

## 7.1 Research Questions

This study was guided by three primary research questions aligned with the proposed hypothesis. The subsequent paragraphs provide a detailed reflection on each of these questions.

- **RQ1: How can RAG be effectively integrated with smaller local LLMs to enhance their performance in healthcare applications, achieving results comparable to larger service-based models?**

  Our experiments indicate that RAG substantially improves the relevance and accuracy of responses generated by smaller local LLMs when addressing healthcare-related queries. By coupling these models with a specialized knowledge base, containing curated medical literature, guidelines, and contextual patient data, the LLM's outputs become more domain-specific and less prone to hallucinations. Crucially, this integration minimizes the need for exhaustive parameter scaling. In our evaluation, Phi4, when combined with efficient retrieval layers, approached the accuracy ranges of GPT-4-1106-preview services in open-answer tasks. Optimizing the search and rerank parameters further aligned local LLM performance with larger service-based counterparts. As a result, the synergy between the generative capabilities of a smaller LLM and a robust RAG subsystem proved to be a viable way to match, or at least closely approximate, the outputs of more extensive commercial APIs in medical question-answer scenarios.

- **RQ2: How do medical-specialized LLMs perform compared to general-purpose LLMs in addressing healthcare-specific tasks?**

In tasks requiring domain-specific knowledge and understanding of medical terminology, medical-specialized large language models have been observed to perform effectively [90]. However, in RAG settings, this study identified limitations in medical local LLMs, such as the Meditron 7B model, that consistently failed to generate responses, likely due to gaps in its architectural design or retrieval mechanisms, resulting in null outputs. Similarly, MedLLaMA 2 demonstrated an inability to follow prompts, producing responses misaligned with evaluation criteria, thus rendering it ineffective for healthcare-specific RAG tasks. In contrast, general-purpose models like GPT-4-1106-preview or Phi4, when paired with robust RAG frameworks, achieved high accuracy, showcasing their ability to integrate retrieved knowledge effectively for complex medical tasks. While medical-specialized LLMs can excel in static, domain-specific tasks, their performance in dynamic, retrieval-dependent applications like RAG is limited, especially when compared to well-optimized general-purpose models.

- **RQ3: Do local LLMs offer significant reductions in monetary costs and energy consumption compared to large, service-based LLMs when used in healthcare settings?**

  The cost and energy analyses confirm that local LLMs, especially Phi4, can be run at substantially lower monetary and energy overhead than the trillion-parameter, service-based solutions. Under high-volume scenarios (e.g., thousands of daily patient queries), the total cost of API-based models such as GPT-4 scales sharply. Local deployment became more economical over time given consistent usage. From an energy standpoint, measured joules per inference were markedly lower for local models. In real-world healthcare environments, where budget constraints and strict data privacy regulations are critical, these values can be pivotal. Hence, for many use cases requiring high throughput, local solutions not only reduce long-term operational costs but also enable stricter control over data security, aligning well with standard privacy mandates.

The results of this study provide evidence in support of the hypothesis that the integration of RAG with local, smaller-scale language models can achieve performance levels comparable to larger, service-based LLMs in healthcare applications. Throughout tasks such as multiple-choice QA and open-ended questions, local models, exemplified by Phi4, under optimized RAG configurations, demonstrated competitive accuracy compared to larger models such as GPT-4o. Moreover, the ability of these smaller models to effectively process extensive retrieval outputs, despite having significantly fewer parameters, underscores their suitability for addressing complex, domain-specific tasks. In addition to performance, the study underscores the considerable advantages offered by local models in terms of monetary and energy efficiency, as well as data privacy. Using local infrastructure, these models eliminate the dependency on external APIs, reducing operational costs associated with service-based solutions. Furthermore, their smaller-scale architecture inherently consumes less energy, aligning with sustainable computational practices. Critically, the use of local models ensures that sensitive healthcare data remains within secure and localized environments.

## 7.2 Main Contributions

- **Demonstration of Feasibility for Local LLMs**

  The dissertation provides evidence that smaller, local language models, when combined with retrieval-augmented generation pipelines, can closely approximate, and in some cases match, the performance of larger, service-based models on healthcare-related tasks. This finding underscores the viability of on-premises AI solutions where data privacy and compliance considerations are paramount.

- **Comprehensive Cost and Energy Assessment**

  The dissertation analyzes the monetary and energy implications of deploying local versus service-based LLMs. The findings highlight significant trade-offs and demonstrate the potential for substantial cost savings and reduced energy consumption when organizations adopt local hosting strategies, especially for high-volume use cases.

- **Guidance on Parameter Tuning and Trade-Offs**

  Through experimentation with search and rerank parameters of our RAG pipeline, the research identifies diminishing returns beyond certain thresholds, providing recommendations for balancing accuracy with resource expenditure, crucial for scalable, real-world applications.

- **Foundational Insights for Healthcare Applications**

  Recognizing the stringent regulatory and ethical frameworks surrounding patient data, this dissertation offers a roadmap for leveraging local LLMs to ensure robust privacy protections without excessively compromising on performance. This contributes to safer and more compliant AI implementations in medical and clinical contexts.

## 7.3 Future Work

This section presents possible directions for research and development that emerged during the course of this work. Although the results obtained have demonstrated the potential of smaller language models, some limitations and open questions indicate the need for further investigation.

This study focused on a specific set of models, so it is advisable to conduct experiments with a broader scope of architectures, including recently launched models and leaner or more robust variants. In this way, it will be possible to identify more precisely which local architectures can present the best relationship between performance, operating cost and suitability for the healthcare sector.

Given the relevance of specialized domains in medical practice, further fine-tuning of local models is suggested, employing even more detailed medical datasets (anonymized clinical records and specialized academic libraries). Moreover, pre-training smaller architectures with high-quality data and medical

curation may be a promising way of improving the adherence of these models to clinical protocols, standardized terminologies and regulatory guidelines.

Although RAG approach has demonstrated benefits in the quality and relevance of responses, there remains a need to evaluate other RAG parameters, such as document indexing variations. Further studies can clarify whether alternative configurations can increase accuracy, reduce response time or improve generalization capacity in more complex clinical scenarios.

The energy measurements mentioned were based on estimations, due to the lack of information on deployment infrastructures and LLMS service parameters, so further explorations on more accurate values could be pursued. Such improvements could provide more reliable and robust estimates of the environmental impact of different inference configurations.

# Bibliography


[1] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics, 2024.

[2] Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S. Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Chenyu You, Xian Wu, Yefeng Zheng, Lei Clifton, Zheng Li, Jiebo Luo, and David A. Clifton. A survey of large language models in medicine: Progress, application, and challenge, 2024.

[3] Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[6] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[9] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022.

[10] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[11] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[12] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

[13] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[14] Kai Lv, Yuqing Yang, Tengxiao Liu, Qinghui Gao, Qipeng Guo, and Xipeng Qiu. Full parameter fine-tuning for large language models with limited resources. *arXiv preprint arXiv:2306.09782*, 2023.

[15] Michael R Zhang, Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. Using large language models for hyperparameter optimization. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[16] Christophe Tribes, Sacha Benarroch-Lelong, Peng Lu, and Ivan Kobyzev. Hyperparameter optimization for large language model instruction-tuning. *arXiv preprint arXiv:2312.00949*, 2023.

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[18] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[19] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[20] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[21] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[22] Yuki Sonoda, Ryo Kurokawa, Yuta Nakamura, Jun Kanzawa, Mariko Kurokawa, Yuji Ohizumi, Wataru Gonoi, and Osamu Abe. Diagnostic performances of gpt-4o, claude 3 opus, and gemini 1.5 pro in "diagnosis please" cases. *Japanese journal of radiology*, 42(11):1231–1235, 2024.

[23] Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, Nasif Zaman, Prithul Sarker, Andrew G Lee, and Alireza Tavakkoli. Google's ai chatbot "bard": a side-by-side comparison with chatgpt and its utilization in ophthalmology. *Eye*, 38(4):642–645, 2024.

[24] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022.

[25] Sebastian Joseph, Kathryn Kazanas, Keziah Reina, Vishnesh J Ramanathan, Wei Xu, Byron C Wallace, and Junyi Jessy Li. Multilingual simplification of medical texts. *arXiv preprint arXiv:2305.12532*, 2023.

[26] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[27] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[30] Lisa C Adams, Daniel Truhn, Felix Busch, Felix Dorfner, Jawed Nawabi, Marcus R Makowski, and Keno K Bressem. Llama 3 challenges proprietary state-of-the-art large language models in radiology board–style examination questions. *Radiology*, 312(2):e241191, 2024.

[31] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[32] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling (2021). *URL https://arxiv.org/abs/2103.10360*, 94, 2022.

[33] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[34] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems, 2023.

[35] Ankit Pal and Malaikannan Sankarasubbu. Gemini goes to med school: exploring the capabilities of multimodal large language models on medical challenge problems & hallucinations. *arXiv preprint arXiv:2402.07023*, 2024.

[36] Pooja Mishra, Rutuja Bhujbal, and Tushar Singh. Exploring the capabilities of large language model mistral large (mistral) on medical challenge problems & hallucinations. *International Research Journal of Innovations in Engineering and Technology*, 8(5):156, 2024.

[37] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[38] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[39] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

[40] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[41] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[42] Lavender Yao Jiang, Xujin Chris Liu, Nima Pour Nejatian, Mustafa Nasir-Moin, Duo Wang, Anas Abidin, Kevin Eaton, Howard Antony Riina, Ilya Laufer, Paawan Punjabi, et al. Health system-scale language models are all-purpose prediction engines. *Nature*, 619(7969):357–362, 2023.

[43] Sultan Alrowili and Vijay Shanker. Large biomedical question answering models with albert and electra. In *CLEF (Working Notes)*, pages 213–220, 2021.

[44] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[45] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*, 2022.

[46] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*, 2019.

[47] Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*, 2021.

[48] Qiuhao Lu, Dejing Dou, and Thien Nguyen. Clinicalt5: A generative language model for clinical text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5436–5443, 2022.

[49] Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651, 2023.

[50] Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323, 2022.

[51] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409, 2022.

[52] A Venigalla, Jonathan Frankle, and M Carbin. Biomedlm: a domain-specific large language model for biomedical text. *MosaicML. Accessed: Dec*, 23(3):2, 2022.

[53] Weihao Gao, Zhuo Deng, Zhiyuan Niu, Fuju Rong, Chucheng Chen, Zheng Gong, Wenze Zhang, Daimin Xiao, Fang Li, Zhenjie Cao, et al. Ophglm: Training an ophthalmology large language-and-vision assistant based on instructions and dialogue. *arXiv preprint arXiv:2306.12174*, 2023.

[54] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, et al. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540*, 2022.

[55] Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language model for medical research and healthcare. *NPJ digital medicine*, 6(1):210, 2023.

[56] Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Linlin Huang, Qian Wang, and Dinggang Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*, 2023.

[57] Yirong Chen, Zhenyu Wang, Xiaofen Xing, Zhipei Xu, Kai Fang, Junhong Wang, Sihang Li, Jieling Wu, Qi Liu, Xiangmin Xu, et al. Bianque: Balancing the questioning and suggestion ability of health llms with multi-turn health conversations polished by chatgpt. *arXiv preprint arXiv:2310.15896*, 2023.

[58] Guangyu Wang, Guoxing Yang, Zongxin Du, Longjun Fan, and Xiaohu Li. Clinicalgpt: large language models finetuned with diverse medical data and comprehensive evaluation. *arXiv preprint arXiv:2306.09968*, 2023.

[59] Qichen Ye, Junling Liu, Dading Chong, Peilin Zhou, Yining Hua, Fenglin Liu, Meng Cao, Ziming Wang, Xuxin Cheng, Zhu Lei, et al. Qilin-med: Multi-stage knowledge injection advanced medical large language model. *arXiv preprint arXiv:2310.09089*, 2023.

[60] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

[61] Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*, 2023.

[62] Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*, 2023.

[63] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023.

[64] Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442*, 2023.

[65] Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*, 2023.

[66] Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. Alpacare: Instruction-tuned large language models for medical application. *arXiv preprint arXiv:2310.14558*, 2023.

[67] Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19368–19376, 2024.

[68] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045, 2024.

[69] Ofir Ben Shoham and Nadav Rappoport. Cpllm: Clinical prediction with large language models. *PLOS Digital Health*, 3(12):e0000680, 2024.

[70] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

[71] Malaikannan Sankarasubbu Ankit Pal and Malaikannan Sankarasubbu. Openbiollms: Advancing open-source large language models for healthcare and life sciences, 2024.

[72] Jinhang Wei, Linlin Zhuo, Xiangzheng Fu, XiangXiang Zeng, Li Wang, Quan Zou, and Dongsheng Cao. Drugrealign: a multisource prompt framework for drug repurposing based on large language models. *BMC biology*, 22(1):226, 2024.

[73] Augustin Toma, Patrick R Lawler, Jimmy Ba, Rahul G Krishnan, Barry B Rubin, and Bo Wang. Clinical camel: An open expert-level medical language model with dialogue-based knowledge encoding. *arXiv preprint arXiv:2305.12031*, 2023.

[74] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature Medicine*, pages 1–8, 2025.

[75] Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Yash Dalmia, Jure Leskovec, Cyril Zakka, Eduardo Pontes Reis, and Pranav Rajpurkar. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367. PMLR, 2023.

[76] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36, 2024.

[77] Stephanie L Hyland, Shruthi Bannur, Kenza Bouzid, Daniel C Castro, Mercy Ranjit, Anton Schwaighofer, Fernando Pérez-García, Valentina Salvatelli, Shaury Srivastav, Anja Thieme, et al.

Maira-1: A specialised large multimodal model for radiology report generation. *arXiv preprint arXiv:2311.13668*, 2023.

[78] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards generalist foundation model for radiology. *arXiv preprint arXiv:2308.02463*, 2023.

[79] Lin Yang, Shawn Xu, Andrew Sellergren, Timo Kohlberger, Yuchen Zhou, Ira Ktena, Atilla Kiraly, Faruk Ahmed, Farhad Hormozdiari, Tiam Jaroensri, et al. Advancing multimodal medical capabilities of gemini. *arXiv preprint arXiv:2405.03162*, 2024.

[80] Valentin Liévin, Christoffer Egeberg Hother, Andreas Geert Motzfeldt, and Ole Winther. Can large language models reason about medical questions? *Patterns*, 5(3), 2024.

[81] Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, et al. Deid-gpt: Zero-shot medical text de-identification by gpt-4. *arXiv preprint arXiv:2303.11032*, 2023.

[82] Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*, 2023.

[83] Yanjun Gao, Ruizhe Li, John Caskey, Dmitriy Dligach, Timothy Miller, Matthew M Churpek, and Majid Afshar. Leveraging a medical knowledge graph into large language models for diagnosis prediction. *arXiv preprint arXiv:2308.14321*, 2023.

[84] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*, 2023.

[85] Wenqi Shi, Yuchen Zhuang, Yuanda Zhu, Henry Iwinski, Michael Wattenbarger, and May Dongmei Wang. Retrieval-augmented large language models for adolescent idiopathic scoliosis patients in shared decision-making. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–10, 2023.

[86] Jaewoong Kim and Moohong Min. From rag to qa-rag: Integrating generative ai for pharmaceutical regulatory compliance process. *arXiv preprint arXiv:2402.01717*, 2024.

[87] Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068, 2024.

[88] Dyke Ferber, Isabella C Wiest, Georg Wölflein, Matthias P Ebert, Gernot Beutel, Jan-Niklas Eckardt, Daniel Truhn, Christoph Springfeld, Dirk Jäger, and Jakob Nikolas Kather. Gpt-4 for information retrieval and comparison of medical oncology guidelines. *NEJM AI*, 1(6):AIcs2300235, 2024.

[89] Hui Feng, Francesco Ronzano, Jude LaFleur, Matthew Garber, Rodrigo de Oliveira, Kathryn Rough, Katharine Roth, Jay Nanavati, Khaldoun Zine El Abidine, and Christina Mack. Evaluation of large language model performance on the biomedical language understanding and reasoning benchmark: Comparative study. *medRxiv*, pages 2024–05, 2024.

[90] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. Meditron-70b: Scaling medical pretraining for large language models, 2023.

[91] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024.

[92] Brian Ondov, Kush Attal, and Dina Demner-Fushman. A survey of automated methods for biomedical text simplification. *Journal of the American Medical Informatics Association*, 29(11):1976–1988, 2022.

[93] Qiao Jin, Zifeng Wang, Charalampos S Floudas, Fangyuan Chen, Changlin Gong, Dara Bracken-Clarke, Elisabetta Xue, Yifan Yang, Jimeng Sun, and Zhiyong Lu. Matching patients to clinical trials with large language models. *ArXiv*, 2023.

[94] Huachuan Qiu, Anqi Li, Lizhi Ma, and Zhenzhong Lan. Psychat: A client-centric dialogue system for mental health support. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 2979–2984. IEEE, 2024.

[95] Tin Lai, Yukun Shi, Zicong Du, Jiajie Wu, Ken Fu, Yichao Dou, and Ziqi Wang. Psy-llm: Scaling up global mental health psychological services with ai-based large language models. *arXiv preprint arXiv:2307.11991*, 2023.

[96] Zilin Ma, Yiyang Mei, and Zhaoyuan Su. Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. In *AMIA Annual Symposium Proceedings*, volume 2023, page 1105. American Medical Informatics Association, 2023.

[97] Zhiyao Ren, Yibing Zhan, Baosheng Yu, Liang Ding, and Dacheng Tao. Healthcare copilot: Eliciting the power of general llms for medical consultation. *arXiv preprint arXiv:2402.13408*, 2024.

[98] Zihao Zhao, Sheng Wang, Jinchen Gu, Yitao Zhu, Lanzhuju Mei, Zixu Zhuang, Zhiming Cui, Qian Wang, and Dinggang Shen. Chatcad+: Towards a universal and reliable interactive cad using llms. *IEEE Transactions on Medical Imaging*, 2024.

[99] Justin Peacock, Andrea Austin, Marina Shapiro, Alexis Battista, and Anita Samuel. Accelerating medical education with chatgpt: an implementation guide. *MedEdPublish*, 13, 2023.

[100] Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, Fan Zhang, Tim Strother, Chunjong Park, Elahe Vedadi, et al. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*, 2024.

[101] Alaa Abd-Alrazaq, Rawan AlSaad, Dari Alhuwail, Arfan Ahmed, Padraig Mark Healy, Syed Latifi, Sarah Aziz, Rafat Damseh, Sadam Alabed Alrazak, Javaid Sheikh, et al. Large language models in medical education: opportunities, challenges, and future directions. *JMIR Medical Education*, 9(1):e48291, 2023.

[102] Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. Plm-icd: Automatic icd coding with pretrained language models. *arXiv preprint arXiv:2207.05289*, 2022.

[103] Hanyin Wang, Chufan Gao, Christopher Dantona, Bryan Hull, and Jimeng Sun. Drg-llama: tuning llama model to predict diagnosis-related group for hospitalized patients. *npj Digital Medicine*, 7(1):16, 2024.

[104] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Anthony B Costa, Mona G Flores, et al. A large language model for electronic health records. *NPJ digital medicine*, 5(1):194, 2022.

[105] Iker García-Ferrero, Rodrigo Agerri, Aitziber Atutxa Salazar, Elena Cabrio, Iker de la Iglesia, Alberto Lavelli, Bernardo Magnini, Benjamin Molinet, Johana Ramirez-Romero, German Rigau, et al. Medical mt5: an open-source multilingual text-to-text llm for the medical domain. *arXiv preprint arXiv:2404.07613*, 2024.

[106] Xidong Wang, Nuo Chen, Junyin Chen, Yan Hu, Yidong Wang, Xiangbo Wu, Anningzhe Gao, Xiang Wan, Haizhou Li, and Benyou Wang. Apollo: Lightweight multilingual medical llms towards democratizing medical ai to 6b people. *arXiv preprint arXiv:2403.03640*, 2024.

[107] Sara Pieri, Sahal Shaji Mullappilly, Fahad Shahbaz Khan, Rao Muhammad Anwer, Salman Khan, Timothy Baldwin, and Hisham Cholakkal. Bimedix: Bilingual medical mixture of experts llm. *arXiv preprint arXiv:2402.13253*, 2024.

[108] Yue Guo, Wei Qiu, Gondy Leroy, Sheng Wang, and Trevor Cohen. Retrieval augmentation of large language models for lay language generation. *Journal of Biomedical Informatics*, 149:104580, 2024.

[109] Masoud Moghani, Lars Doorenbos, William Chung-Ho Panitch, Sean Huver, Mahdi Azizian, Ken Goldberg, and Animesh Garg. Sufia: Language-guided augmented dexterity for robotic surgical assistants. *arXiv preprint arXiv:2405.05226*, 2024.

[110] Huan Xu, Jinlin Wu, Guanglin Cao, Zhen Lei, Zhen Chen, and Hongbin Liu. Enhancing surgical robots with embodied intelligence for autonomous ultrasound scanning. *arXiv preprint arXiv:2405.00461*, 2024.

[111] Benjamin D Killeen, Shreayan Chaudhary, Greg Osgood, and Mathias Unberath. Take a shot! natural language control of intelligent robotic x-ray systems in surgery. *International journal of computer assisted radiology and surgery*, pages 1–9, 2024.

[112] Induni N Weerarathna, David Raymond, and Anurag Luharia. Human-robot collaboration for healthcare: A narrative review. *Cureus*, 15(11), 2023.

[113] Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*, 2(2):102–115, 2024.

[114] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew BA McDermott, Irene Y Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12):2176–2182, 2021.

[115] Avishek Choudhury and Onur Asan. Impact of accountability, training, and human factors on the use of artificial intelligence in healthcare: Exploring the perceptions of healthcare practitioners in the us. *Human Factors in Healthcare*, 2:100021, 2022.

[116] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[117] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[118] Jing Miao, Charat Thongprayoon, Supawadee Suppadungsuk, Oscar A Garcia Valencia, and Wisit Cheungpasitporn. Integrating retrieval-augmented generation with large language models in nephrology: advancing practical applications. *Medicina*, 60(3):445, 2024.

[119] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine, 2024.

[120] Mohammad Alkhalaf, Ping Yu, Mengyang Yin, and Chao Deng. Applying generative ai with retrieval augmented generation to summarize and extract key clinical information from electronic health records. *Journal of Biomedical Informatics*, page 104662, 2024.

[121] Mingyu Jin, Qinkai Yu, Dong Shu, Chong Zhang, Lizhou Fan, Wenyue Hua, Suiyuan Zhu, Yanda Meng, Zhenting Wang, Mengnan Du, and Yongfeng Zhang. Health-llm: Personalized retrieval-augmented disease prediction system, 2024.

[122] Qingqing Zhou, Can Liu, Yuchen Duan, Kaijie Sun, Yu Li, Hongxing Kan, Zongyun Gu, Jianhua Shu, and Jili Hu. Gastrobot: a chinese gastrointestinal disease chatbot based on the retrieval-augmented generation. *Frontiers in Medicine*, 11, 2024.

[123] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms, 2024.

[124] Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 752–762, 2024.

[125] Taojun Hu and Xiao-Hua Zhou. Unveiling llm evaluation focused on metrics: Challenges and solutions, 2024.

[126] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study, 2023.

[127] Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 606–615, 2024.

[128] Tyler Griggs, Xiaoxuan Liu, Jiaxiang Yu, Doyoung Kim, Wei-Lin Chiang, Alvin Cheung, and Ion Stoica. Mélange: Cost efficient large language model serving by exploiting gpu heterogeneity, 2024.

[129] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

[130] Abi Aryan, Aakash Kumar Nain, Andrew McMahon, Lucas Augusto Meyer, and Harpreet Singh Sahota. The costly dilemma: generalization, evaluation and cost-optimal deployment of large language models. *arXiv preprint arXiv:2308.08061*, 2023.

[131] Micah Musser. A cost analysis of generative language models and influence operations. *arXiv preprint arXiv:2308.03740*, 2023.

[132] Chansung Park, Juyong Jiang, Fan Wang, Sayak Paul, Jing Tang, and Sunghun Kim. Llamaduo: Llmops pipeline for seamless migration from service llms to small-scale local llms. *arXiv preprint arXiv:2408.13467*, 2024.

[133] Yuhang Yao, Han Jin, Alay Dilipbhai Shah, Shanshan Han, Zijian Hu, Yide Ran, Dimitris Stripelis, Zhaozhuo Xu, Salman Avestimehr, and Chaoyang He. Scalellm: A resource-frugal llm serving framework by optimizing end-to-end efficiency. *arXiv preprint arXiv:2408.00008*, 2024.

[134] Shivanshu Shekhar, Tanishq Dubey, Koyel Mukherjee, Apoorv Saxena, Atharv Tyagi, and Nishanth Kotla. Towards optimizing the costs of llm usage, 2024.

[135] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference, 2023.

[136] Carlos Baquero. The energy footprint of humans and large language models: Comparing the energy expenditure of people with that of large language models, 2023. Accessed: 2024-10-24.

[137] Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. Towards greener llms: Bringing energy-efficiency to the forefront of llm inference. *arXiv preprint arXiv:2403.20306v1*, 2024.

[138] Joseph McDonald, Baolin Li, Nathan Frey, Devesh Tiwari, Vijay Gadepally, and Siddharth Samsi. Great power, great responsibility: Recommendations for reducing energy for training language models. *arXiv preprint arXiv:2205.09646*, 2022.

[139] 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.

[140] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

[141] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019.

[142] Jae-Won Chung, Jiachen Liu, Zhiyu Wu, Yuxuan Xia, and Mosharaf Chowdhury. ML.ENERGY leaderboard. https://ml.energy/leaderboard, 2023.

[143] Paul Joe Maliakel, Shashikant Ilager, and Ivona Brandic. Investigating energy efficiency and performance trade-offs in llm inference across tasks and dvfs settings, 2025.

[144] Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. Long context rag performance of large language models, 2024.

[145] Nghia Trung Ngo, Chien Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. Comprehensive and practical evaluation of retrieval-augmented generation systems for medical question answering, 2024.

[146] Jinqiang Wang, Huansheng Ning, Yi Peng, Qikai Wei, Daniel Tesfai, Wenwei Mao, Tao Zhu, and Runhe Huang. A survey on large language models from general purpose to medical applications: Datasets, methodologies, and evaluations. *arXiv preprint arXiv:2406.10303*, 2024.

[147] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*, 2024.