UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Comparative Analysis of Open Source Large Language Models

Author:
Victor FAYOS I PÉREZ

Supervisor:
Dr. Daniel ORTIZ MARTINEZ
Julio César ARPÍREZ VEGA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc in Fundamental Principles of Data Science

Comparative Analysis of Open Source Large Language Models

by Victor FAYOS I PÉREZ

This study investigates the potential of using smaller, locally hosted language models (LLMs) to perform specific tasks traditionally handled by large LLMs, such as OpenAI's Chat-GPT 3.5. With the growing integration of LLMs in corporate environments, concerns over costs, data privacy, and security have become prominent. By focusing on question answering and text summarization tasks, we compare the performance of several smaller models, including Flan T5 XXL, Phi 3 Mini, and Yi 1.5, against Chat-GPT 3.5. As the two experiments show, one on question answering and the second one on text summarization, this tasks can be done by the tested models at the same level than the state of the art Chat-GPT 3.5. Concluding that depending the use intended for the LLM one of the different models could best fit as the variety in the response structure and verbosity highly depends on the model selected.

Acknowledgements

I wish to extend my gratitude to my teacher, Dr. Daniel Ortiz, for his tutoring and guidance throughout the course of this research. His expertise, feedback, and support have been pivotal in the successful completion of this study. I am thankfull to my company *Samsung Zhilabs* for their provision of resources and the flexibility afforded to me during this project. Their commitment to supporting my professional development has been of great help in facilitating the progress and realization of this research. Additionally, I would like to express my heartfelt appreciation to my loved ones for their support, encouragement, and understanding. Their patience and belief in my abilities have been a constant source of motivation and strength.

Thank you to all who have contributed to the fruition of this work.

Contents

Chapter 1

Introduction to the problem

1.1 Motivation

This project aims to address some of the primary challenges associated with integrating a chatbot or a large language model (LLM) into a product. The two main problems are cost and privacy, and our proposed solution effectively mitigates these issues.

Many companies currently rely on third-party LLMs like Chat-GPT, Gemini, and Bert for their applications. While this approach is easy to implement, requiring minimal computational resources and offering quick deployment, it has two significant drawbacks: high costs and potential data security risks, despite assurances of data protection. Alternatively, training an in-house LLM tailored specifically to a product's needs can yield optimal results but demands substantial time and financial investment.

The approach proposed in this study is a hybrid solution that combines the benefits of both methods. This involves batch processing—where critical requests are handled overnight by a large LLM and refreshed daily—and then using a smaller, locally hosted model to respond to customer inquiries during the day, utilizing the batch responses as part of its input. This strategy ensures that most data remains within the company's network, reducing the risk of data breaches and minimizing the number of calls to expensive, large LLMs. Additionally, the resources required for this solution are minimal, as the proposed models can be hosted on relatively small machines.

1.2 LLMs in the Corporate World

Since the groundbreaking release of OpenAI's Chat-GPT3 in November 2022, Large Language Models (LLMs) have transformed the landscape of artificial intelligence applications across industries. These models, equipped with advanced natural language processing capabilities, have quickly become indispensable tools for companies seeking to enhance customer interactions, streamline operations, and drive innovation.

The introduction of Chat-GPT3 to the general public marked a turning point in the accessibility and awareness of LLMs. Witnessing the potential of these models to automate customer support, personalize recommendations, and generate human-like text, businesses of all sizes began exploring ways to integrate LLMs into their workflows.

In response to the demand for tailored solutions, major technology companies such as Google, X (formerly known as SpaceX), and Meta have developed their own proprietary LLMs. Google introduced Gemini, optimized for search and information retrieval tasks, while X deployed Grok for enhancing internal communication and knowledge management. Meta's adoption of Llama underscores its commitment to leveraging advanced AI for social media platforms, focusing on content moderation and user engagement strategies.

For smaller enterprises or those without the resources to develop and maintain their own LLMs, the availability of API services from these tech giants has been instrumental. By accessing LLM capabilities via APIs, companies can harness the power of these models without the upfront costs associated with infrastructure and ongoing maintenance. This approach democratizes access to advanced AI technologies, empowering businesses to innovate and compete in a rapidly evolving digital landscape.

While using these APIs might seem like the best option to integrate this new technology into products, there are several drawbacks to consider. These include not only the financial costs but also significant privacy and security concerns associated with sending data to an external party. In this paper, we focus on the possibility of delegating simple tasks to local models, thereby reducing dependence on external models.

1.3 Challenges with LLMs

While the integration of LLMs promises significant benefits for corporate operations, several critical challenges must be addressed to maximize their effectiveness and mitigate risks.

1.3.1 Financial Costs and Resource Allocation

One of the foremost challenges in deploying LLMs within large enterprises is the substantial financial investment required throughout the model lifecycle. From initial training to continuous optimization and deployment, the costs associated with acquiring high-performance computing resources, maintaining robust data centers, and employing skilled AI engineers can be prohibitive.

Even for organizations opting to utilize pre-trained LLMs through API services, cost considerations remain paramount. Pricing structures typically hinge on the volume of data processed by the model, encompassing both incoming queries and outgoing responses. While this pay-per-use model may be manageable for smaller entities with moderate transaction volumes, it can quickly escalate for multinational corporations handling vast datasets and frequent model interactions.

Strategic resource allocation is crucial to managing these financial implications effectively. Companies must weigh the benefits of deploying LLMs against the operational costs and consider alternative approaches, such as hybrid models combining in-house capabilities with external API services, to optimize expenditure and maximize return on investment.

1.3.2 Data Privacy and Security Concerns

LLMs derive their efficacy from extensive training on diverse datasets, including internet content and user interactions. While this training enhances their ability to comprehend and generate human-like text, it also raises significant data privacy concerns for organizations entrusted with sensitive customer information.

API providers emphasize rigorous data protection measures, including encryption, secure data handling protocols, and compliance with regulatory standards such as GDPR[7] and CCPA[6]. However, the inherent risks associated with handling large volumes of data—potentially containing personally identifiable information (PII)—persist. There is a tangible risk that LLMs could inadvertently learn from and expose confidential data, compromising user privacy and undermining trust in organizational data governance practices.

Mitigating these risks necessitates a multifaceted approach. Companies must implement robust data anonymization techniques, enforce strict access controls, and conduct regular security audits to safeguard against potential breaches. Moreover, fostering transparency with stakeholders about data usage policies and practices is essential to building and maintaining trust in the ethical deployment of LLM technologies.

1.4 Proposed Solution

The solution proposed in this study advocates for the adoption of smaller, specialized language models hosted on compact hardware setups. This strategic shift aims to mitigate the substantial financial investments associated with maintaining large-scale clusters required by massive LLMs and with the millions of tokens send to the APIs. By opting for smaller models, companies can significantly reduce operational costs, as the maintenance and resource requirements are considerably lower. This approach not only economizes on infrastructure but also addresses critical concerns surrounding data privacy and security.

Hosting smaller models within company premises ensures that sensitive data and user inputs remain within controlled environments, minimizing the risk of unauthorized access or data breaches. Unlike larger models that may involve sending data to external servers for processing, local hosting enhances confidentiality and compliance with stringent data protection regulations.

The primary objective of this paper is to conduct a comparative analysis of three leading open-source small LLMs. Specifically, the study focuses on evaluating their performance in two distinct tasks: Question Answering and Text Summarization. By benchmarking these smaller models against the industry benchmark GPT-3.5[14], the research aims to assess their efficacy and limitations in handling complex natural language processing tasks.

By exploring the capabilities of these smaller models, the study seeks to provide valuable insights into their suitability for practical deployment in corporate environments. This comparative analysis will contribute to understanding the trade-offs between model size, computational efficiency, and task-specific performance, thereby informing decision-making processes for organizations looking to leverage advanced AI technologies effectively.

Chapter 2

State of the art

In the rapidly evolving field of LLMs, recent models have significantly pushed the boundaries of what is possible with machine learning models. This three models are the ones that will be put to test: Phi 3, Yi 1.5, and Flan T5 XXL. Each of these models represents a significant leap in performance and capability, addressing various challenges in NLP through novel methodologies and architectures.

2.1 Preliminaries

In order to understand the following models we need to understand the concepts explained below so the description and the functioning of the model is clear.

2.1.1 Transformer Architecture

The transformer model, introduced in the groundbreaking paper *Attention is All You Need*[21] by Vaswani et al. (2017), is a neural network architecture that has fundamentally transformed natural language processing. Unlike previous sequential models, transformers process all words in a sentence simultaneously through a mechanism called self-attention. This allows the model to weigh the importance of each word in a context, irrespective of its position, facilitating parallelization and enhancing efficiency.

Key components of the transformer architecture include:

- **Self-Attention Mechanism**: This mechanism enables the model to focus on different parts of the input sequence when producing each word in the output sequence, capturing dependencies regardless of their distance apart.
- **Multi-Head Attention**: Multiple self-attention layers work in parallel, allowing the model to consider various aspects of each word's relationship to other words.
- **Positional Encoding**: Since transformers do not inherently process sequences in order, positional encodings are added to give the model information about the position of words in the sequence.

Understanding the transformer architecture is crucial because it underpins the structure and functionality of most modern LLMs, enabling them to handle and generate human language effectively.

2.1.2 Scaling Laws of Training a Model

Scaling laws [9] in machine learning describe how the performance of models improves as their size and the amount of training data increase. For LLMs, these

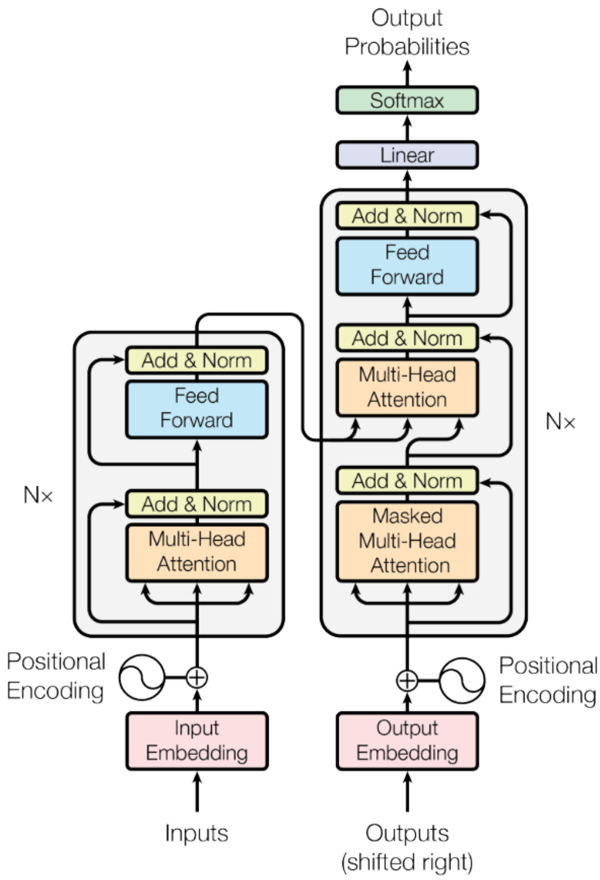

FIGURE 2.1: Diagram of Transformer Architecture

laws highlight the relationship between model size, training data, computational resources, and model performance.

Key points include:

- **Model Size**: Increasing the number of parameters in an LLM generally leads to better performance, as larger models can capture more complex patterns in the data.
- **Training Data**: More extensive and diverse datasets improve the model's ability to generalize, reducing overfitting and enhancing its application across various tasks.
- **Computational Resources**: Training larger models requires significant computational power and efficient algorithms to manage resources effectively and ensure scalability.

Understanding these scaling laws helps in appreciating the trade-offs and challenges involved in developing state-of-the-art LLMs, emphasizing the importance of balancing model complexity with practical constraints.

2.1.3 Prompt Engineering

Prompt engineering is the process of designing and optimizing input prompts to guide LLMs in generating desired outputs. As LLMs are versatile and capable of

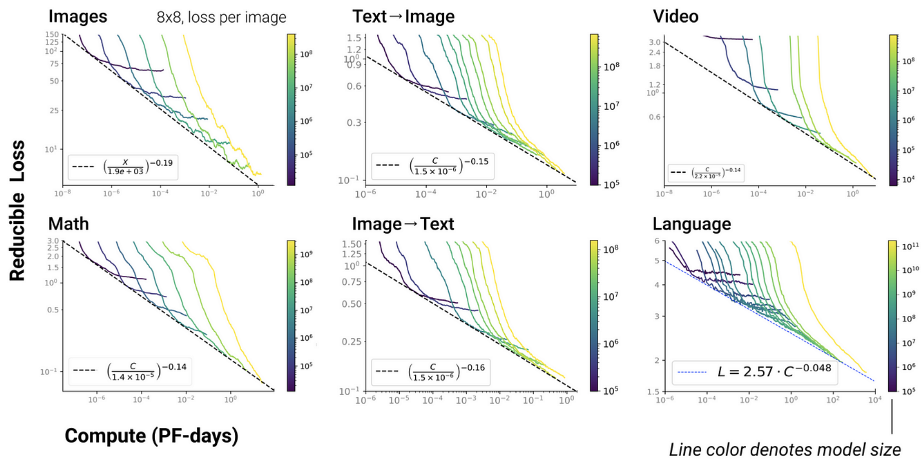

FIGURE 2.2: Example of Scaling Laws by number of parameters in Model Training

performing a wide range of tasks, effective prompt engineering can significantly enhance their performance and utility.

Key aspects include:

- **Clarity and Specificity**: Crafting prompts that clearly and specifically outline the desired task or question helps the model generate more accurate and relevant responses.
- **Context and Examples**: Providing context or examples within the prompt can guide the model to understand the nuances of the task and produce more targeted outputs.
- **Iterative Refinement**: Experimenting with different prompt formulations and iteratively refining them based on the model's responses can lead to optimal performance for specific applications.

Mastering prompt engineering enables users to leverage the full potential of LLMs, tailoring their capabilities to meet diverse and specific needs effectively.

2.1.4 Ethical Considerations

The deployment of LLMs brings significant ethical challenges that must be addressed to ensure their responsible use. Key ethical considerations include:

- **Bias and Fairness**: LLMs trained on large datasets can inadvertently learn and perpetuate biases present in the data, leading to unfair treatment of certain groups. It is essential to recognize and mitigate these biases to promote fairness.
- **Misinformation**: The ability of LLMs to generate coherent and convincing text raises concerns about their potential to spread misinformation or produce harmful content.
- **Privacy**: Training LLMs on vast amounts of data, which may include sensitive information, raises privacy issues. Ensuring data is anonymized and used responsibly is critical to protecting individuals' privacy.

- **Accountability**: As LLMs are increasingly integrated into various applications, establishing clear accountability for their outputs and decisions becomes vital to address misuse and unintended consequences.

Addressing these ethical considerations fosters a balanced view of LLMs' potential and pitfalls, encouraging their development and use in a socially responsible manner.

2.2 Phi 3

Phi 3[1] is an advanced transformer-based model developed by Microsoft, emphasizing efficiency and scalability. The model's training incorporates novel techniques to enhance learning and inference capabilities. Additionally, it is aligned for robustness, safety, and optimized for chat formats. Phi 3 is available in three versions: **phi-3-mini** (3.8B parameters), **phi-3-small** (7B parameters), and **phi-3-medium** (14B parameters). This study focuses on the **phi-3-mini**, particularly the variant with the larger context length.

2.2.1 Technical Specifications

The **phi-3-mini** model features a transformer decoder architecture with a context length of **128k**. Its block structure is based on Llama-2[20], and it shares the same tokenizer, with a vocabulary size of 32,064. This compatibility allows all packages developed for Llama-2 to be directly adapted to Phi 3. The model has 3,072 hidden dimensions, 32 heads, and 32 layers, and it was trained using bfloat16 for a total of 3.3T tokens. As mentioned earlier, the model is pre-fine-tuned for chat, utilizing the template: *<|user|>/n Question <|end|>/n <|assistant|>*.

The **phi-3-mini** is also capable of running on small devices, such as cell phones. When quantized to 4-bits, it occupies only 1.8GB of memory.

2.2.2 Training Methodology

The training process follows the methods outlined in "Textbooks Are All You Need"[10], utilizing high-quality training data to improve the performance of small language models and surpass standard scaling laws. This approach enables the **phi-3-mini** to achieve performance levels comparable to larger models like GPT-3.5 or Mixtral[4], despite having fewer parameters.

The data used for training was a mix of heavily filtered publicly available web data and synthetic LLM-generated data. Pre-training occurred in two distinct phases: the first phase focused on learning general knowledge and language understanding from a broad range of internet data, while the second phase emphasized logical reasoning and specialized skills using synthetic data and additional internet data.

Post-training consisted of two stages. The first stage involved supervised fine-tuning with English examples and highly curated data across diverse domains, such as math, coding, reasoning, conversation, and safety. The second stage was direct preference optimization, which covered chat format data, reasoning, and responsible AI, ensuring the model rejects unwanted behaviors and interacts safely.

2.2.3 Safety

Microsoft ensures the safety of the Phi 3 model by adhering to its responsible AI principles[3]. The model's alignment process includes post-training adjustments,

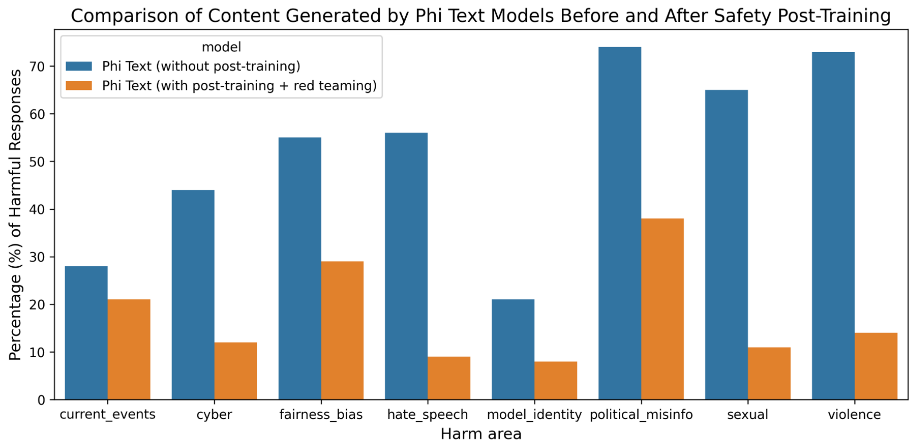

FIGURE 2.3: Plot comparing the harmful responses of Phi 3 mini model with and without post training

red-teaming, automated testing, and evaluations across numerous responsible AI harm categories, significantly reducing the likelihood of harmful responses.

2.3 Yi 1.5

The Yi [2] model series is developed by *01.AI*. In creating these models, the primary focus was on the joint scaling of model size and data quality. The models are designed to be handled by consumer-grade hardware, such as the RTX 4090 with its 24GB of memory. This consideration led the company to create three different models with 34B, 9B, and 6B parameters. This study will focus on the smallest model, which has 6B parameters.

2.3.1 Technical Specifications

Yi models use a modified version of the classic decoder-only Transformer architecture, with the codebase adapted from Llama's implementation and several modifications. The first significant modification is the use of Grouped-Query Attention[5] across all models, including the 6B parameter model. To support large context windows, up to 200k, the model employs Rotary Position Embeddings[19]. Training the model with 1-2B tokens has proven sufficient to achieve low loss over context lengths ranging from 4k to 200k.

The Yi-1.5 model with 6B parameters features 4,096 hidden dimensions, 32 heads, and 32 layers. It was trained with a sequence length of 4,096 and on 3.1T tokens.

2.3.2 Training Methodology

The training methodology prioritizes data quality over quantity. The dataset comprises 10k multi-turn instruction-response dialog pairs, each constructed and refined through multiple iterations and user feedback.

Several techniques were employed to enhance prompt distribution selection, response formatting, and chain-of-thought formatting. For prompt distribution, compound instructions were developed and progressively evolved to increase complexity. Response formatting follows an extended style from LIMA[22], structured in an introduction-body-conclusion format, with the body often presented as bullet

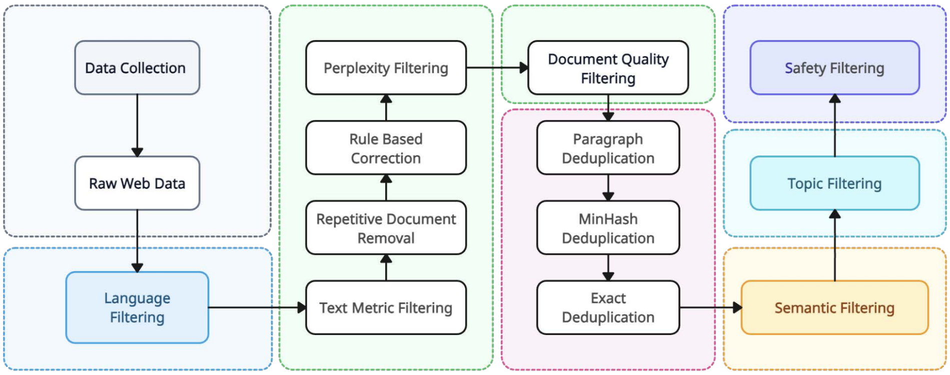

FIGURE 2.4: Yi 1.5 pretraining data cleaning pipeline

points. For chain-of-thought data formatting, higher-level abstractions were formulated before addressing the original, concrete questions. Additional efforts were made to reduce hallucinations and repetition, ensuring that responses do not contain memorized knowledge.

To ensure diversity, a wide spectrum of open-source prompts was included, covering areas such as question answering, creative writing, dialogue, reasoning, mathematics, coding, safety, and bilingual capabilities, among others.

2.3.3 Safety

Yi models incorporate safety measures both in pre-training and alignment. During pre-training, a set of filters based on heuristic rules, keyword matching, and learned classifiers was used to remove texts containing personal identifiers, private data, and reduce sexual, violent, and extremist content. In the alignment phase, a comprehensive safety taxonomy was developed to address a broad spectrum of potential concerns. Additionally, a series of attack prompts were simulated to improve the model's resilience against malicious use.

2.4 Flan-T5

Flan-T5[18] is an advanced language model developed by Google, building upon the T5 (Text-To-Text Transfer Transformer[16]) architecture. The focus of Flan-T5 is on fine-tuning large language models with a diverse set of instruction prompts to improve their ability to follow human instructions accurately. The model is designed to operate efficiently on various hardware setups, including high-performance GPUs like the NVIDIA A100 with 40GB of memory. Flan-T5 comes in multiple versions, including small, base, large, XL, and XXL models. This study will focus on the XXL version of Flan-T5, which has approximately 11 billion parameters.

2.4.1 Technical Specifications

Flan-T5 utilizes the standard T5 architecture, which is an encoder-decoder model. The XXL version of Flan-T5 has approximately 11 billion parameters, with the encoder and decoder each having 24 layers. The model features 4,096 hidden dimensions and 64 attention heads. The model employs relative positional encodings to manage token positions within the input sequences. The model was pre-trained on the C4 dataset and then fine-tuned on a diverse set of instruction-following tasks to enhance its generalization capabilities across various NLP tasks.

2.4.2 Training Methodology

The training methodology for Flan-T5 emphasizes the importance of instruction tuning. The model was initially pre-trained on the C4 dataset, which comprises extensive and diverse text data from the internet. After pre-training, Flan-T5 underwent a rigorous fine-tuning process using a mixture of instruction-following tasks. The instruction-tuning dataset included thousands of tasks ranging from translation, summarization, and question answering to more complex reasoning and problem-solving tasks.

During fine-tuning, several techniques were employed to improve the model's ability to follow human instructions. These included the use of task-specific prompts, response formatting, and chain-of-thought prompting. Task-specific prompts were designed to guide the model towards the desired outputs for various tasks. Response formatting was standardized to ensure clarity and consistency, often employing structured formats such as bullet points or step-by-step instructions. Chain-of-thought prompting encouraged the model to break down complex tasks into simpler, logical steps.

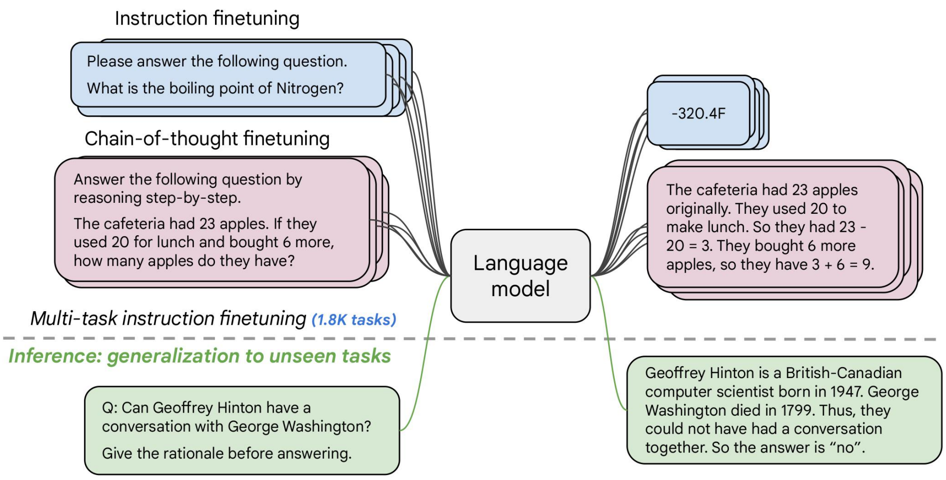

FIGURE 2.5: Diagram of the Finetuning of Flan T5 XXL

To ensure high-quality output, special attention was given to reducing hallucinations and ensuring the accuracy of generated responses. This was achieved through iterative refinement and the incorporation of user feedback into the training process.

2.4.3 Safety

Safety is a critical aspect of Flan-T5's development. During pre-training, content filtering mechanisms were employed to remove harmful or inappropriate data from the training corpus. These filters included heuristic rules, keyword matching, and learned classifiers to exclude texts containing personal identifiers, private data, and content that could be considered violent, sexual, or extremist.

In the alignment phase, Flan-T5 was evaluated against a comprehensive safety taxonomy designed to identify and mitigate potential risks. This involved simulating various attack prompts to test the model's resilience against malicious inputs.

The model's responses were also manually reviewed to ensure they adhered to ethical guidelines and minimized harmful outputs.

2.5 Model Comparison

The following table provides a detailed comparison of these three models across various features, including model size and computational efficiency. These comparison offers insights into the capabilities and trade-offs associated with each model.

Benchmark	**Phi-3 3.8B**	**Yi-1.5 6B**	**Flan-T5 XXL**
Model Size (Parameters)	3.8B	6B	11B
Hidden Dimensions	3072	4096	4096
Attention Heads	32	32	64
Layers	32	32	24
Context Length	128k	200k	Variable
Training Tokens	3.3T	3.1T	Not specified
Tokenization	Llama-2 Tokenizer	Custom (Llama-based)	T5 Tokenizer
Pre-training Dataset	Mixed Web Data	Mixed Web Data	C4 Dataset
Fine-tuning	Instruction tuning	Instruction tuning	Instruction tuning
Safety Measures	Filtering and Alignment	Filtering and Alignment	Filtering and Alignment
MMLU Score (5-shot)	68.8%	61.0%	48.6%

TABLE 2.1: Comparison of Phi-3 3.8B, Yi-1.5 6B, and Flan-T5 XXL models.

Chapter 3

Experimental Setup

The range of tasks that a Large Language Model (LLM) can perform is extensive. Generally, larger models tend to deliver better performance across a wider variety of tasks. However, in our experiment, we focus on a specific subset of tasks to be handled by smaller models. Specifically, we aim to evaluate their performance on two tasks: question answering with data and text summarization.

3.1 Experiment Setup

The experiments have been conducted in a machine with Intel(R) Xeon(R) Silver 4410Y with 64Gb of RAM and a RTX A4000, GDDR6 with 16GB of memory. All the models have been downloaded from hugginface web page and deployed via CPU. This decision was made due to the large RAM memory available, so the models didn't have to be quatized for the testing and so the results could reflect the optimal conditions for each model. All the experiments code is available in the github repository provided [8].

3.2 Question Answering

Question answering (QA) systems play a pivotal role in natural language processing by enabling models to comprehend and respond to queries posed by users based on textual input. The primary objective of this experiment is to evaluate the efficacy of a model in accurately answering questions derived from a given text or document.

QA systems serve as indispensable tools across various domains, offering users a streamlined method to extract specific information from extensive texts, reports, or datasets. This capability is particularly valuable in scenarios where users need quick access to relevant insights without manually sifting through large volumes of information.

The experimental framework involves assessing how effectively the model processes and interprets natural language queries, ranging from factual inquiries to more nuanced requests requiring contextual understanding. By measuring the model's accuracy and efficiency in generating precise responses, this experiment aims to validate its capability to enhance information retrieval processes across different applications.

3.2.1 Stanford Question Answering Dataset (SQuAD)

The Stanford Question Answering Dataset (SQuAD)[17] is a widely-used reading comprehension dataset, consisting of questions derived from a set of Wikipedia articles. Each question is designed to have its answer contained within a specific segment of the corresponding article. The dataset's questions and answers are created by human annotators through crowdsourcing, resulting in a diverse collection of question-answer pairs compared to other question-answering datasets. SQuAD contains 107,785 question-answer pairs spread across 536 articles, with 87,599 pairs used for training, 10,570 pairs reserved for validation and 9,616 for test.

The dataset features are as follows:

- **id**: Identifier of the question.
- **title**: Title of the question.
- **context**: Text containing all the information necessary to answer the question.
- **question**: The question to be answered.
- **answers**: A dictionary with two items regarding the answer.
  - **text**: The text of the ground truth response.
  - **answer_start**: The character position in the context where the answer starts.

In our experiment, we decided to adopt two specific measures for evaluation:

Focus on Text Answers: We will consider only the textual content of the answers, disregarding the *answer_start* feature. This decision is driven by practical considerations, prioritizing the correctness of the answer over whether the answer was directly extracted from the text or inferred through the model's prior knowledge. Given that the models being tested are relatively small, their inherent knowledge base is limited, allowing us to safely assume that the answers are derived from the provided context.

Subset of Training Samples: We will utilize a subset of 1,000 samples from the training set for evaluation. This subset size is chosen to ensure a representative assessment of each model's performance on the task, balancing computational efficiency with sufficient coverage to yield meaningful insights.

These measures are intended to provide a robust evaluation framework that emphasizes practical applicability and fairness in assessing the models' question-answering capabilities. By focusing on textual correctness and using a representative sample size, we aim to derive conclusions that are both relevant and reliable for real-world applications.

3.2.2 Experiment Configuration

The experiment was conducted sequentially, focusing on one model and one query at a time. Each query was carefully crafted through prompt engineering, involving testing approximately six different versions to optimize performance for the task. Central to the approach was the segmentation of the query into two distinct blocks: the context block and the data block.

The context block serves as the foundation of knowledge, containing the information where the answer to the query is expected to reside. By instructing the model to focus on this contextual information, provided solely through the prompt, we aimed to direct its attention effectively.

Conversely, the data block comprises the specific question posed to the model. Placing this block after the context block was a strategic decision, proven to enhance performance in comparison to alternative arrangements.

Moreover, we employed a prompting technique known as Directional Stimulus Prompting [reference]. This technique guides the model towards generating responses of a particular type. In our experiment, the objective was to solicit concise responses for direct comparison with ground truth answers, facilitating evaluation and bench-marking.

Finally the prompt that was used has the following structure:

Provide the shortest answer. Data: ***context***. Question: ***question***

3.3 Text Summarization

Text summarization is a critical task in natural language processing aimed at condensing large volumes of information into concise and coherent summaries. The goal of this experiment is to assess the ability of a model to generate accurate and informative summaries from extensive textual inputs.

Text summarization serves as a fundamental tool across various domains, enabling users to quickly grasp essential information from lengthy documents, articles, or datasets. This capability not only enhances efficiency in information consumption but also facilitates decision-making processes by providing synthesized insights.

In this experiment, the model's performance is evaluated based on its capacity to distill key points, themes, and arguments from input texts of varying lengths and complexities. The evaluation criteria include assessing the coherence, relevance, and informativeness of the generated summaries compared to human-authored summaries or ground truth references.

3.3.1 WikiHow Dataset

The WikiHow Dataset[11] offers a compact and challenging resource for text summarization, standing in contrast to the widely used CNN/Daily Mail dataset. Each WikiHow article consists of multiple paragraphs, with each paragraph beginning with a summarizing sentence. For the purposes of this dataset, these paragraphs are merged to form a complete article, while the introductory sentences are combined to create the summary.

There are two primary versions of the dataset available for use:

1. **Full Article Summarization**: This version concatenates all the paragraphs of the article and all the bold introductory lines for the summary, resulting in a structure as follows:
    - **Title**: The title of the article.
    - **Headline**: Concatenation of all the bold introductory lines from each paragraph, serving as the reference summary.
    - **Text**: Concatenation of all paragraphs, excluding the bold lines, forming the article to be summarized.
2. **Modular Paragraph Summarization**: This version focuses on individual paragraphs and their respective summaries, structured as follows:
    - **Title**: The title of the article.
    - **Overview**: Introduction to the paragraph.

- **Headline**: The bold introductory line of the paragraph, serving as the reference summary.
- **Text**: The paragraph content, excluding the bold line, to be summarized.

In our experiment, we utilized the more comprehensive version involving full articles. This decision was driven by the anticipated real-world use case where the model is applied to summarize extensive input texts, providing users with concise descriptions of lengthy documents. This approach tests the model's ability to abstract a summary from a large chunk of text effectively.

Due to the higher computational demands of text summarization compared to question answering, we limited the experiment to a subset of 100 samples. This sample size is sufficient to provide a representative assessment of each model's performance while maintaining computational feasibility.

By using the full article version, our evaluation emphasizes the model's capability to handle and summarize larger texts, aligning with practical applications where comprehensive summarization is required. This approach ensures that the results are both relevant and insightful for potential deployment scenarios.

3.3.2 Experiment Configuration

The experiment was conducted methodically, focusing on evaluating one model and one query at a time. Each query was extracted directly from the dataset, as preliminary tests indicated that the specific formulation of the prompt had minimal impact on task performance.

The prompt utilized in the experiment was deliberately straightforward, designed to direct the model to summarize a provided text efficiently. Through iterative testing, it was determined that variations in how the task was framed did not significantly alter the model's performance. Consequently, the finalized prompt structure employed in the experiment was as follows:

Get the most important concepts from this data. Data: ***text***

3.4 Metrics

In this study, we employed automatic evaluation metrics to address these challenges. Specifically, we utilized four widely recognized metrics. These metrics are untrained, meaning they do not require a pre-trained algorithm to generate scores. Moreover, they are generic and language-independent, making them versatile for evaluating models across different languages while maintaining consistent evaluation standards. Although this study is conducted in English, the same evaluation approach can be applied to other languages without altering the evaluation process.

The four metrics selected are: BLEU, precision based, ROUGE, recall based, METEOR precision and recall based, and BERTScore, LLM based.

3.4.1 Bilingual Evaluation Understudy (BLEU)

The Bilingual Evaluation Understudy[15] is a widely recognized metric, frequently cited in the literature on machine translation and was introduced in a seminal paper in 2002. The metric compares a candidate translation to one or more reference translations, yielding a score between 0 and 1, with values closer to 1 indicating higher quality translations.

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3.1}$$

Where:

- BP (Brevity Penalty) is defined as:

$$BP = \begin{Bmatrix} 1, & \text{if } c > r \\ \exp(1 - \frac{r}{c}), & \text{if } c \leq r \end{Bmatrix} \tag{3.2}$$

  where c is the length of the candidate translation and r is the effective reference corpus length.

- p_n is the modified n-gram precision, calculated as:

$$p_n = \frac{\sum_{C \in \text{Candidates}} \sum_{n\text{-gram} \in C} \min(\text{count}(n\text{-gram}), \max_{\text{reference} \in \text{References}} \text{count}(n\text{-gram}))}{\sum_{C \in \text{Candidates}} \sum_{n\text{-gram} \in C} \text{count}(n\text{-gram})} \tag{3.3}$$

- w_n is the weight for n-gram precision, usually uniform and summing up to 1 (e.g., for BLEU-4, $w_1 = w_2 = w_3 = w_4 = 0.25$).

- N is the maximum n-gram length (e.g., $N = 4$ for BLEU-4).

BLEU-1, BLEU-2, BLEU-3, and BLEU-4 are commonly reported variants of the BLEU score, where the numbers 1, 2, 3, and 4 represent the precision of n-gram sequences considered in the calculation. Each variant measures the precision for sequences of different lengths, providing a comprehensive assessment of translation accuracy.

The metric as seen in the formulas incorporates a "Brevity Penalty" to address the issue of overly short translations. If the candidate translation is shorter than the reference, the BLEU score is penalized accordingly, ensuring that the metric does not favor truncated outputs over complete ones.

This metric could be applied either sentence to sentence or document to document. In the study it was decided the option of sentence to sentence in order to get one score for each prediction, this caused the score to be almost binary.

3.4.2 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

The Recall-Oriented Understudy for Gisting Evaluation[13] was introduced in a seminal paper in 2004. It is a widely used metric in the literature on text summarization, designed to calculate the syntactic overlap between candidate and reference summaries (or other text pieces).

Several variants of ROUGE are commonly reported, including ROUGE-N, ROUGE-L, and ROUGE-S. ROUGE-N measures the overlap of n-grams between the candidate and reference texts. ROUGE-L evaluates the longest co-occurring sequence of n-grams, providing insight into the sequential similarity between the candidate and reference. Finally, ROUGE-S calculates the overlap of skip-bigrams (any pair of words in their sentence order) between the texts.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{References}} \sum_{\text{n-gram} \in S} \min(\text{count}_{\text{match}}(\text{n-gram}), \text{count}_{\text{candidate}}(\text{n-gram}))}{\sum_{S \in \text{References}} \sum_{\text{n-gram} \in S} \text{count}(\text{n-gram})} \tag{3.4}$$

where:

- $\text{count}_{\text{match}}(\text{n-gram})$ is the number of n-grams co-occurring in the candidate and reference summaries.
- $\text{count}_{\text{candidate}}(\text{n-gram})$ is the number of n-grams in the candidate summary.
- $\text{count}(\text{n-gram})$ is the number of n-grams in the reference summary.

$$\text{ROUGE-L} = F_{\text{score}} = \frac{(1+\beta^2)\cdot \text{R}_{\text{LCS}} \cdot \text{P}_{\text{LCS}}}{\text{R}_{\text{LCS}} + \beta^2 \cdot \text{P}_{\text{LCS}}} \tag{3.5}$$

where:

- $\text{R}_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{|Y|}$ is the recall of the LCS.
- $\text{P}_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{|X|}$ is the precision of the LCS.
- $\text{LCS}(X,Y)$ is the length of the longest common subsequence between candidate summary X and reference summary Y.
- β is a parameter that determines the relative importance of precision and recall ($\beta = 1$ typically means equal weight to precision and recall).

$$\text{ROUGE-S} = \frac{\sum_{S\in\text{References}} \sum_{\text{skip-bigram}\in S} \min(\text{count}_{\text{match}}(\text{skip-bigram}), \text{count}_{\text{candidate}}(\text{skip-bigram}))}{\sum_{S\in\text{References}} \sum_{\text{skip-bigram}\in S} \text{count}(\text{skip-bigram})} \tag{3.6}$$

where:

- $\text{count}_{\text{match}}(\text{skip-bigram})$ is the number of skip-bigrams co-occurring in the candidate and reference summaries.
- $\text{count}_{\text{candidate}}(\text{skip-bigram})$ is the number of skip-bigrams in the candidate summary.
- $\text{count}(\text{skip-bigram})$ is the number of skip-bigrams in the reference summary.

3.4.3 Metric for Evaluation of Translation with Explicit ORdering (METEOR)

The Metric for Evaluation of Translation with Explicit ORdering[12] metric is versatile and can be used to evaluate outputs across a range of tasks, including Machine Translation, Text Summarization, and Image Captioning. However, it is most commonly referenced in the literature on machine translation due to its high correlation with human judgment.

METEOR offers several advantages over the BLEU metric, which primarily focuses on precision in comparing generated text to ground truth. In contrast, METEOR calculates the harmonic mean of unigram precision and recall, with greater weight given to recall. This emphasis on recall allows for a more comprehensive understanding of how much of the ground truth content is captured in the generated output.

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - \text{Penalty}) \tag{3.7}$$

where:

- F_{mean} is the harmonic mean of precision (P) and recall (R), calculated as:

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P} \tag{3.8}$$

- Penalty is a penalty factor calculated as:

$$\text{Penalty} = 0.5 \cdot \left(\frac{ch}{m}\right)^3 \tag{3.9}$$

  where ch is the number of chunks and m is the number of matches.

- Precision (P) and Recall (R) are defined as:

$$P = \frac{\text{number of matches}}{\text{number of words in candidate}} \tag{3.10}$$

$$R = \frac{\text{number of matches}}{\text{number of words in reference}} \tag{3.11}$$

  A match can be an exact match, stemmed match, synonym match, or paraphrase match.

METEOR incorporates a "Chunk Penalty" that evaluates the overlap of not just unigrams but also chunks (consecutive words), thereby accounting for the order of words in the candidate text.

In METEOR's calculation, ch represents the number of chunks in the candidate that also occur in the reference, and m refers to the unigrams in the candidate sentence. The final METEOR score (M) is derived by multiplying the factor p with the F-score, incorporating both precision and recall elements.

3.4.4 BERTScore

As the name suggests, BERTScore leverages BERT for evaluating the quality of text generated by Natural Language Generation (NLG) systems. Introduced in a 2020 paper, BERTScore differs from traditional metrics that primarily rely on token or phrasal level syntactic overlaps between hypothesis and reference texts. Instead, BERTScore captures the semantic aspect by utilizing contextualized embeddings generated by the BERT model.

The BERTScore evaluation process begins by obtaining contextualized word embeddings for both the reference (ground truth) and candidate (generated by the NLG system) text pieces. It then calculates the cosine similarity between each word in the reference and candidate texts, identifying semantically similar words. Precision, Recall, and F-score are subsequently calculated based on these similarity scores.

An additional feature of BERTScore is the incorporation of "importance weights," which assign more weight to words deemed significant and unique to the text. Inverse Document Frequencies (IDF) weights are used for this purpose. The intuition behind this approach is that more common words across documents are less critical to the specific document, resulting in lower IDF values. Conversely, unique words have higher IDF values, indicating their importance. This weighting mechanism is flexible and open to further exploration.

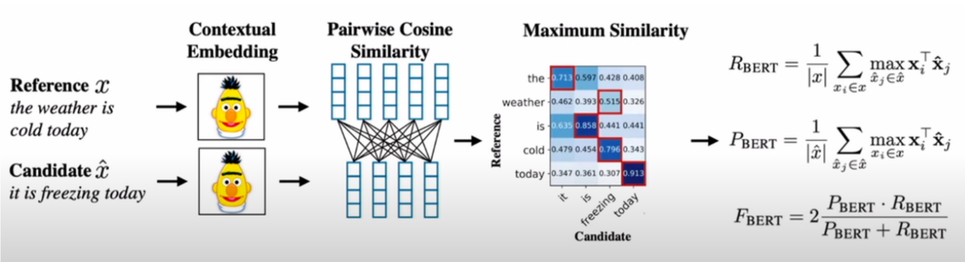

FIGURE 3.1: BERTScore Pipeline

Chapter 4

Results

Evaluating these models can be challenging. One of the most effective methods for assessing the performance of a large language model (LLM) is through human evaluation. Although human evaluation is highly regarded for its nuanced insights, it is time-consuming and prohibitively expensive when conducted on a large scale. Additionally, this method lacks reproducibility due to inherent human biases.

The results of both experiments will be analyzed using a combination of quantitative and qualitative methods. Quantitative analysis will be performed through the metrics we defined earlier, providing objective measurements of model performance. Qualitative analysis, on the other hand, will focus on the practical application and contextual relevance of the models in their intended deployment environments.

Additionally, to provide a benchmark and an approximate idea of the performance of each model, the same samples have been tested against *Chat-GPT 3.5*, the most widely used model today. This comparative analysis will help contextualize the capabilities of the smaller models relative to a leading large language model.

Qualitative evaluation is inherently subjective and can vary significantly depending on the specific requirements and constraints of the final product. This aspect of the analysis emphasizes the practical usability and effectiveness of the models in real-world scenarios, ensuring that the models not only perform well in controlled tests but also meet the nuanced needs of their operational context.

4.1 Question Answering

4.1.1 Quantitative Analysis

In this task the prediction and the ground truth of the elements are small, this makes it easier to evaluate the model due to the responses need to be almost identical to consider them good. As we can see in almost all the metrics the metrics are binary, or 1 or 0.

The most relevant metrics for this experiment are **BLEU**, **ROUGE**, and **METEOR**. These three scores are sensitive to the accuracy and recall of words between the ground truth and the prediction. The plots indicates that the best-performing model across all scores is *Flan-T5 XXL*, followed by *Chat-GPT 3.5*, *Phi 3 Mini*, and finally *Yi 1.5*.

Regarding the **BLEU** score, we can infer that the responses given by *Flan-T5 XXL* are extremely similar to the ground truth, as the **BLEU** score is computed with short sequences of words, where even a single mistake can drastically reduce the score.

Focusing on the **ROUGE** score, the difference between *Flan-T5 XXL* and *Chat-GPT 3.5* is not significant. As this metric is recall-based, it suggests that while *Chat-GPT 3.5*'s responses contain the necessary information, they also include additional words that penalize the **BLEU** score.

Lastly, examining the **METEOR** score, *Flan-T5 XXL* still emerges as the best model, but the other three models exhibit similar performance. This reinforces that *Flan-T5 XXL* provides responses that closely match the ground truth, whereas the other models, although containing the correct information, have additional words that lower their scores.

4.1.2 Qualitative Analysis

The following two examples illustrate the types of responses generated by each model and explain why the scores are distributed as they are. In both examples, all four models provide the correct information. Across the dataset, only a few responses were incorrect or lacked sufficient information. However, the models exhibit notably different styles in their responses.

The *Flan T5 XXL* model consistently provides concise answers, often matching the ground truth responses. As the largest model among those tested (excluding *Chat-GPT 3.5*), it is likely that during its training phase, the model learned extensive factual knowledge, which sometimes comes at the expense of the naturalness in its responses. This characteristic makes the *Flan T5 XXL* particularly suitable for applications where users, especially those with a technical profile, need precise and quick answers from a given text.

The *Phi 3 Mini* and *Yi 1.5* models provide answers that often resemble a formal conversation between individuals. While these responses may result in lower metrics, they are conceptually correct. These models are smaller in size and have been trained with carefully selected data, which means they have minimal prior knowledge but a strong understanding of phrase structure and information extraction. This makes them suitable for broader public use, as their responses appear less robotic yet remain accurate.

Example 1: YouTube

- **Context:** Google does not provide detailed figures for YouTube's running costs, and YouTube's revenues in 2007 were noted as "not material" in a regulatory filing. In June 2008, a Forbes magazine article projected the 2008 revenue at $200 million, noting progress in advertising sales. In January 2012, it was estimated that visitors to YouTube spent an average of 15 minutes a day on the site, in contrast to the four or five hours a day spent by a typical U.S. citizen watching television. In 2012, YouTube's revenue from its ads program was estimated at 3.7 billion. In 2013 it nearly doubled and estimated to hit 5.6 billion dollars according to eMarketer, others estimated 4.7 billion.

- **Question:** What is the average time an American watches TV in day?

- **Ground Truth Answer:** four or five hours

- **Flan T5 XXL Response:** four or five hours

- **Chat-GPT 3.5 Response:** Four or five hours.

- **Phi 3 Mini Response:** Americans watch TV for 4 or 5 hours a day.

- **Yi 1.5 Response:** The average time an American watches TV in a day is 4-5 hours.

Example 2: Raleigh, North Carolina

- **Context:** The region also experiences occasional periods of drought, during which the city sometimes has restricted water use by residents. During the late summer and early fall, Raleigh can experience hurricanes. In 1996, Hurricane Fran caused severe damage in the Raleigh area, mostly from falling trees. The most recent hurricane to have a considerable effect on the area was Isabel in 2003. Tornadoes also have on occasion affected the city of Raleigh most notably the November 28, 1988 tornado which occurred in the early morning hours and rated an F4 on the Fujita Tornado Scale and affected Northwestern portions of the city. Also the April 16, 2011 F3 Tornado which affected portions of downtown and North east Raleigh and the suburb of Holly Springs.
- **Question:** What does the area experience?
- **Ground Truth Answer:** drought
- **Flan T5 XXL Response:** drought
- **Chat-GPT 3.5 Response:** hurricanes, droughts, tornadoes
- **Phi 3 Mini Response:** The area experiences droughts, hurricanes, and tornadoes.
- **Yi 1.5 Response:** The area experiences occasional periods of drought, restricted water use by residents, hurricanes (especially in late summer and early fall), and tornadoes.

Model	BLEU	ROUGE	METEOR
Example 1: YouTube			
Flan T5 XXL	1.00	1.00	0.99
Chat-GPT 3.5	0.00	1.00	0.74
Phi 3 Mini	0.00	0.29	0.86
Yi 1.5	0.00	0.11	0.00
Example 2: Raleigh, North Carolina			
Flan T5 XXL	1.00	1.00	1.00
Chat-GPT 3.5	0.00	0.76	0.00
Phi 3 Mini	0.00	0.25	0.00
Yi 1.5	0.00	0.09	0.00

TABLE 4.1: Evaluation Scores for Example Responses

4.2 Text Summarization

4.2.1 Quantitative Analysis

Unlike the first task, this one involves generating lengthy outputs. Evaluating such responses poses a challenge for traditional scoring metrics to accurately assess their quality. In this experiment, *BERTScore* plays a crucial role because it compares how well the information in the reference and generated answers align. This metric helps identify whether the models under test include all necessary information or if there are discrepancies compared to the ground truth.

The **BERTScore** metric reveals several key insights. Firstly, by examining precision, we notice that *Chat-GPT 3.5* only outperforms the *Yi 1.5* model, while only the *Flan T5 XXL* model delivers more accurate answers compared to the reference model. When we consider the recall plot, it becomes evident that all the models tested generate responses that are similar in nature, as in the plot we can see overlapping distributions. These responses are well-structured, encompassing all the information from the ground truth, but differ in length, resulting in lower precision.

In contrast, the *Flan T5 XXL* and *Chat-GPT 3.5* models provide more precise responses, meaning they tend to be more concise and focused, containing only the essential information. This approach results in higher precision scores as they avoid extraneous details. Consequently, these models are better suited for applications where brevity and accuracy are paramount, while *Phi 3 Mini* and *Yi 1.5* may be more appropriate for contexts where a more elaborate and comprehensive response is preferred, as tend to be more structured and detailed.

4.2.2 Qualitative Analysis

The following example illustrates how the trends observed in the quantitative analysis are reflected in the types of responses each model delivers. In the dataset, the ground truth summaries are the highlighted words of each paragraph, meaning the targets are small and concise.

The *Flan T5 XXL* model approximates the ground truth most accurately, as its responses are short and concise, containing only the necessary information. This makes it the best model for quick and straightforward analysis. Following this, the other three models, while structured similarly, present their responses in a list format of the most important features of the text. Upon closer inspection, we see that although all the information from the ground truth is included in these lists, there is also additional information. This extra information can be beneficial in some cases but may also detract from the task's focus in others.

Example: How to Clean Gunk from Your Cat's Eyes

- **Text:** You should bring your cat to the vet if you notice the gunk is not going away and your cat's eyes appear very red, as this could be a symptom of conjunctivitis, or pink eye. You should also bring your cat to the vet if the gunk looks green or yellow and is sticky or smelly, as this could be a sign of a bacterial infection.If you notice your cat is constantly pawing her eye or scratching at it, you should bring her to the vet. She may also tear up a lot and her eyes may appear sensitive to light. These are all possible signs of a cornea or inner eye issue. , Allow your vet to examine your cat's eyes. She may take a sample of the gunk to test it for bacteria and observe if your cat's eyes appear very red, inflamed, irritated, or sensitive to light., Your vet will recommend specific treatments based on the type of infection or issue your cat has. Many eye issues can be treated with antibiotic eye ointment or drops that you give your cat for a set period of time. Often, with the right treatment, eye issues heal well in cats.If your cat has conjunctivitis or corneal disorders, your vet may recommend an antibiotic ointment. She may also suggest keeping your cat's eyes clean on a regular basis. If your cat has a blocked tear duct, she may use plain water or a saline solution to flush out the duct.

- **Ground Truth Answer:** Check your cat for signs of an eye infection., Get a diagnosis from the vet., Discuss your treatment options.
- **Flan T5 XXL Response:** Bring your cat to the vet if the gunk is not going away. Allow your vet to examine your cat's eyes. Ask your vet for treatment recommendations.
- **Chat-GPT 3.5 Response:** - Symptoms of eye issues in cats: gunk not going away, red eyes, green or yellow gunk, constant pawing or scratching, tearing up, sensitivity to light - Possible issues: conjunctivitis, bacterial infection, cornea or inner eye issue - Vet examination: sample testing, observation of symptoms - Treatments: antibiotic eye ointment or drops, keeping eyes clean, flushing blocked tear duct - With the right treatment, eye issues in cats can heal well.
- **Phi 3 Mini Response:** 1. Conjunctivitis (Pink Eye) 2. Bacterial eye infection 3. Green or yellow eye discharge 4. Eye irritation and redness 5. Excessive tearing 6. Sensitivity to light 7. Eye examination by a vet 8. Bacterial culture test 9. Antibiotic eye ointment or drops 10. Cleaning the cat's eyes regularly 11. Flushing blocked tear ducts with water or saline solution 12. Possible corneal or inner eye issue 13. Veterinary treatment recommendations based on infection type 14. Healing of eye issues with proper treatment.
- **Yi 1.5 Response:** 1. **Symptom of Conjunctivitis or Pink Eye**: Gunk not going away and very red eyes. 2. **Bacterial Infection Symptoms**: Gunk appears green or yellow, is sticky or smelly. 3. **Signs of Inner Eye Issues**: Constant pawing, scratching, excessive tearing, and eyes appear sensitive to light. 4. **Importance of Vet Examination**: Allow the vet to examine the eyes, taking a sample of the gunk to test for bacteria and observing symptoms of redness, inflammation, irritation, and sensitivity to light. 5. **Treatment Recommendations**: Based on the issue, treatments may include antibiotic eye ointment or drops, with regular administration for a set period. For conjunctivitis or corneal disorders, an antibiotic ointment may be recommended, along with regular eye cleaning. For blocked tear ducts, water or saline solution may be suggested.

Model	BERTScore Precision	BERTScore Recall	BERTScore F1-Score
Flan T5 XXL	0.91	0.91	0.91
Chat-GPT 3.5	0.83	0.88	0.85
Phi 3 Mini	0.82	0.88	0.85
Yi 1.5	0.81	0.88	0.84

TABLE 4.2: Evaluation Scores for Text Summarization

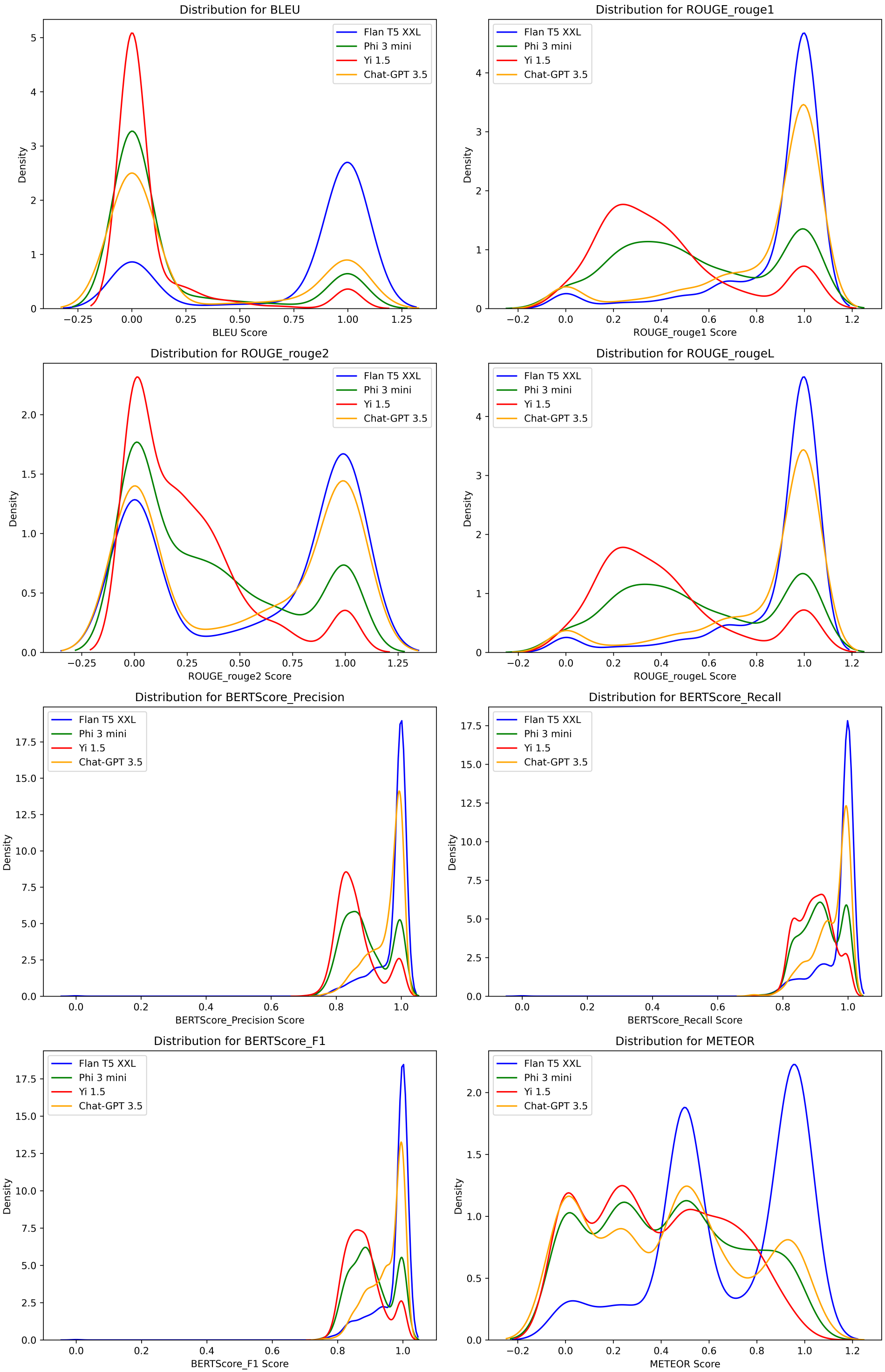

FIGURE 4.1: Question Answering Scores Distributions

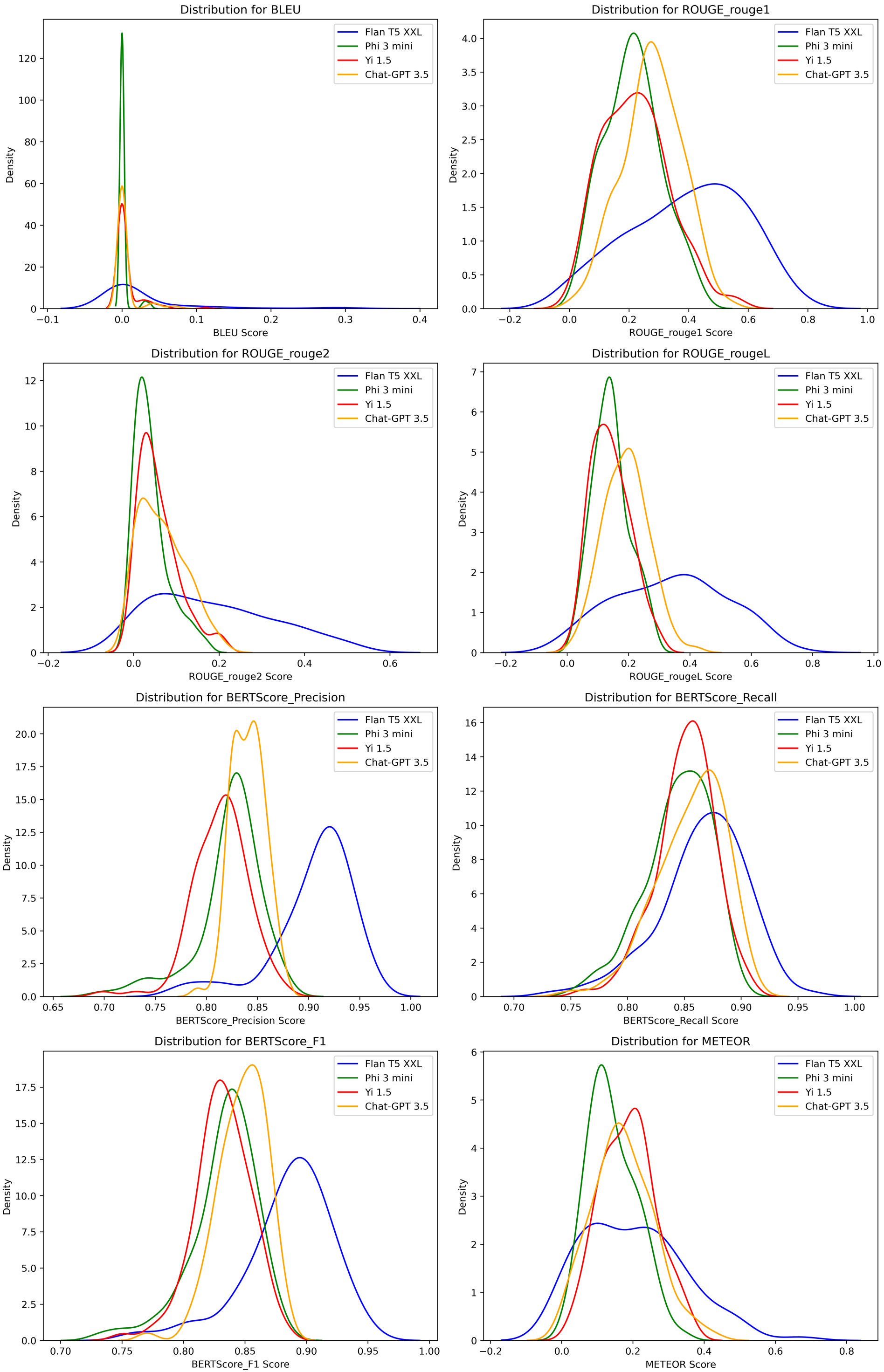

FIGURE 4.2: Text Summarization Scores Distributions

Chapter 5

Conclusions

In this study, we evaluated the performance of several large language models (LLMs) on the tasks of question answering and text summarization. By comparing *Flan T5 XXL*, *Phi 3 Mini*, and *Yi 1.5* to *Chat-GPT 3.5* we aimed to understand how well the models performed against the state of the art model.

The *Flan T5 XXL* model emerged as the most concise, often providing answers that closely match the ground truth of the datasets used. This model, being one of the largest with 11B of parameters, seems to prioritize accuracy over elaboration, making it highly suitable for technical applications requiring precise information retrieval.

In contrast, *Yi 1.5* generates responses that, while structurally sound and informative, tend to be longer and less precise with the ground truth of the datasets. This model is better suited for applications where a conversational tone and comprehensive answers are valued, making it suitable for the general public. Their responses, although correct in content, include more extraneous information, leading to lower precision scores.

The *Phi 3 Mini* model strikes a balance between brevity and detail. As its responses are not as well structured as the ones the model *Yi 1.5*, neither so concise as the ones from *Flan T5 XXL*. It produces accurate answers with essential information, making it suitable for contexts where both accuracy and conciseness are important.

The BERTScore metric, particularly when analyzing precision and recall, highlighted these differences effectively. *Flan T5 XXL* and *Phi 3 Mini* excelled in precision, indicating a focus on essential information without unnecessary elaboration. Conversely, *Yi 1.5* showed higher recall, suggesting its responses contain all necessary information but with added verbosity.

Overall, our findings suggest that substituting large LLMs with smaller, domain-specific models for certain tasks can be done effectively without compromising quality. This approach enables mid-sized and large companies to deploy their own clusters with tailored models, fostering the creation of new, specialized LLM architectures. By adopting this strategy, companies can significantly enhance application performance and mitigate data privacy concerns, as all data remains under the company's control.

Another significant observation is that the quality of data and the feedback provided to the model greatly enhance its comprehension and performance. This suggests that merely increasing a model's computational resources is not the most effective way to improve its capabilities. Instead, the focus should be on meticulously processing the training data and conducting thorough post-training fine-tuning. This approach ensures that the model learns from high-quality, relevant data, leading to more accurate and reliable outputs.

Lastly, this study also paves the way for architectures consisting of multiple domain-specific models. In such a systems, a collection of models, each excelling in a particular task, can collaborate to achieve better overall performance than a single large LLM. This modular strategy not only optimizes resource usage but also allows for more flexible and efficient AI deployment across various applications.

Bibliography

[1] Marah Abdin et al. *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*. 2024. arXiv: 2404.14219 [cs.CL]. URL: https://arxiv.org/abs/2404.14219.

[2] 01. AI et al. *Yi: Open Foundation Models by 01.AI*. 2024. arXiv: 2403.04652 [cs.CL]. URL: https://arxiv.org/abs/2403.04652.

[3] Microsoft AI. *Responsible AI Principles and Approach*. Accessed: 2024-06-29. 2024. URL: https://www.microsoft.com/en-us/ai/principles-and-approach.

[4] Mistral AI. *Mixtral of Experts: A High-Quality Sparse Mixture-of-Experts*. Accessed: 2024-06-29. 2023. URL: https://mistral.ai/news/mixtral-of-experts/.

[5] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*. 2023. arXiv: 2305.13245 [cs.CL]. URL: https://arxiv.org/abs/2305.13245.

[6] *California Consumer Privacy Act (CCPA)*. 2024. URL: https://oag.ca.gov/privacy/ccpa.

[7] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council*. of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). May 4, 2016. URL: https://data.europa.eu/eli/reg/2016/679/oj (visited on 04/13/2023).

[8] Victor Fayos. *Comparative Analysis of Open Source Large Language Models*. https://github.com/vfayosp/comparative-analysis-of-open-source-large-language-models. GitHub repository. 2024. (Visited on 06/29/2024).

[9] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[10] Jared Kaplan et al. "Textbooks Are All You Need". In: *arXiv preprint arXiv:2006.06799* (2020). URL: https://arxiv.org/abs/2006.06799.

[11] Mahnaz Koupaee and William Yang Wang. *WikiHow: A Large Scale Text Summarization Dataset*. 2018. arXiv: 1810.09305 [cs.CL]. URL: https://arxiv.org/abs/1810.09305.

[12] Alon Lavie and Abhaya Agarwal. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics. 2007, pp. 228–231.

[13] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[14] OpenAI. *ChatGPT: OpenAI's GPT-3.5 Language Model*. Accessed: 2023-06-29. 2022. URL: https://openai.com/research/chatgpt.

[15] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics. 2002, pp. 311–318.

[16] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[17] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. arXiv: 1606.05250 [cs.CL]. URL: https://aclanthology.org/D16-1264.

[18] Adam Roberts et al. "Scaling Up Models and Data with `t5x` and `seqio`". In: *arXiv preprint arXiv:2203.17189* (2022). URL: https://arxiv.org/abs/2203.17189.

[19] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023. arXiv: 2104.09864 [cs.CL]. URL: https://arxiv.org/abs/2104.09864.

[20] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv preprint arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971.

[21] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[22] Chunting Zhou et al. *LIMA: Less Is More for Alignment*. 2023. arXiv: 2305.11206 [cs.CL]. URL: https://arxiv.org/abs/2305.11206.