



3

Choosing an LLM for Your Application

In the last chapter, we saw how pivotal it is to properly orchestrate **large language models (LLMs)** and their components within applications. In fact, we saw that not all LLMs are created equal. The next key decision is which LLMs to actually use. Different LLMs may have different architectures, sizes, training data, capabilities, and limitations. Choosing the right LLM for your application is not a trivial decision, as it can have a significant impact on the performance, quality, and cost of your solution.

In this chapter, we will guide you through the process of choosing the right LLM for your application. We will cover the following topics:

- An overview of the most promising LLMs in the market
- The main criteria and tools to use when comparing LLMs
- Trade-offs between size and performance

By the end of this chapter, you should have a clear understanding of how to choose the right LLM for your application and how to use it effectively and responsibly.

The most promising LLMs in the market

The last year has witnessed an unprecedented surge in the research and development of LLMs. Several new models have been released or announced by different organizations, each with its own features and

capabilities. Some of these models are the largest and most advanced ever created, surpassing the previous **state-of-the-art (SOTA)** by orders of magnitude. Others are lighter yet more specialized in specific tasks.

In this chapter, we will review some of the most promising LLMs in the market as of 2024. We will introduce their background, key findings, and main techniques. We will also compare their performance, strengths, and limitations on various benchmarks and tasks. We will also discuss their potential applications, challenges, and implications for the future of AI and society.

Proprietary models

Proprietary LLMs are developed and owned by private companies, and they are not disclosed with code. They are also typically subject to a fee for consumption.

Proprietary models offer a series of advantages, including better support and maintenance as well as safety and alignment. They also tend to outperform open-source models in terms of generalization, because of their complexity and training datasets. On the other hand, they act as a “black box,” meaning that owners do not disclose the source code to developers.

In the next sections, we will cover three of the most popular proprietary LLMs in the market, as of August 2023.

GPT-4

Released in March 2023, GPT-4 is, together with its newly released “cousin” GPT-4 Turbo, one of the latest models developed by **OpenAI**, is among the top performers in the market at the time of writing this book (while OpenAI, as confirmed by its CEO Sam Altman, is already working on GPT-5).

It belongs to the class of **generative pretrained transformer (GPT)** models, a decoder-only transformer-based architecture introduced by OpenAI. The following diagram shows the basic architecture:

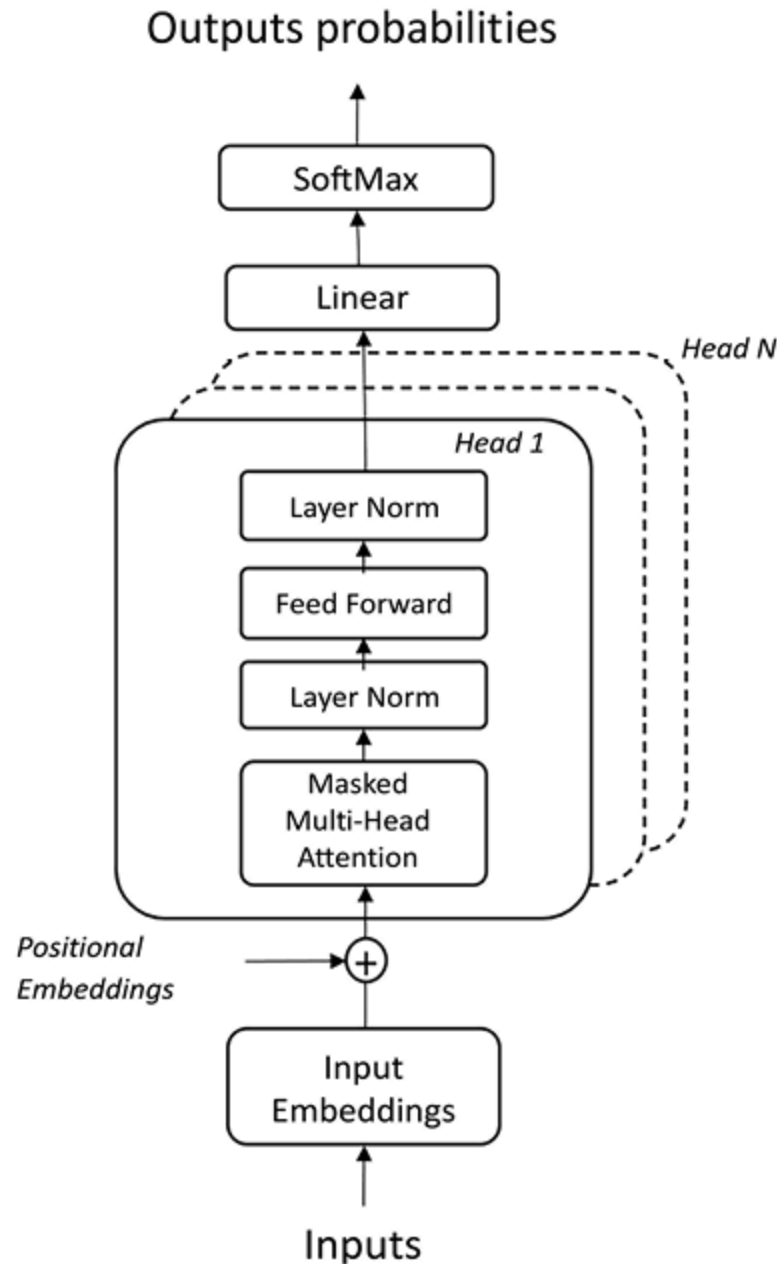


Figure 3.1: High-level architecture of a decoder-only transformer

As you can see from the preceding diagram, the decoder-only architecture still includes the main elements that feature in transformer architecture that we covered in *Chapter 1*, *Positional Embeddings*, *Multi-Head Attention*, and *Feed Forward* layers. However, in this architec-

ture, the model solely comprises a decoder, which is trained to predict the next token in a sequence based on the preceding tokens. Unlike the encoder-decoder architecture, the decoder-only design lacks an explicit encoder for summarizing input information. Instead, the information is implicitly encoded within the hidden state of the decoder, which is updated at each step during the generation process.

Now, we'll look at some of the improvements in GPT-4 over previous versions.

GPT-4, like the previous models in the GPT series, has been trained on both publicly available and OpenAI-licensed datasets (OpenAI didn't disclose the exact composition of the training set).

Additionally, to make the model more aligned with the user's intent, the training process also involved **reinforcement learning from human feedback (RLHF)** training.

Definition

RLHF is a technique that aims at using human feedback as an evaluating metric for LLMs' generated output and then using that feedback to further optimize the model. There are two main steps to achieve that goal:

1. Training a reward model based on human preferences.
2. Optimizing the LLM with respect to the reward model. This step is done via reinforcement learning and it is a type of machine learning paradigm where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on its actions, and its goal is to maximize the cumulative reward over time by



continuously adapting its behavior through trial and error.

With RLHF, thanks to the reward model, the LLM is able to learn from human preferences and be more aligned with users' intents.

As an example, think about ChatGPT. This model integrates various training methods, including unsupervised pretraining, supervised fine-tuning, instruction tuning, and RLHF. The RLHF component involves training the model to predict human preferences by using feedback from human trainers. These trainers review the model's responses and provide ratings or corrections, guiding the model to generate more helpful, accurate, and aligned responses.

For instance, if a language model initially produces an output that is not quite helpful or accurate, human trainers can provide feedback that indicates the preferred output. The model then uses this feedback to adjust its parameters and improve future responses. This process iteratively continues, with the model learning from a series of human judgments to better align with what is considered helpful or appropriate by human standards.

GPT-4 demonstrated outstanding capabilities in commonsense reasoning and analytical skills. It has been benchmarked with SOTA systems, including the **Massive Multitask Language Understanding (MMLU)** we covered in *Chapter 1*. On MMLU, GPT-4 outperformed previous models not only in English, but also in other languages.

The following is an illustration that shows GPT-4's performance on MMLU:

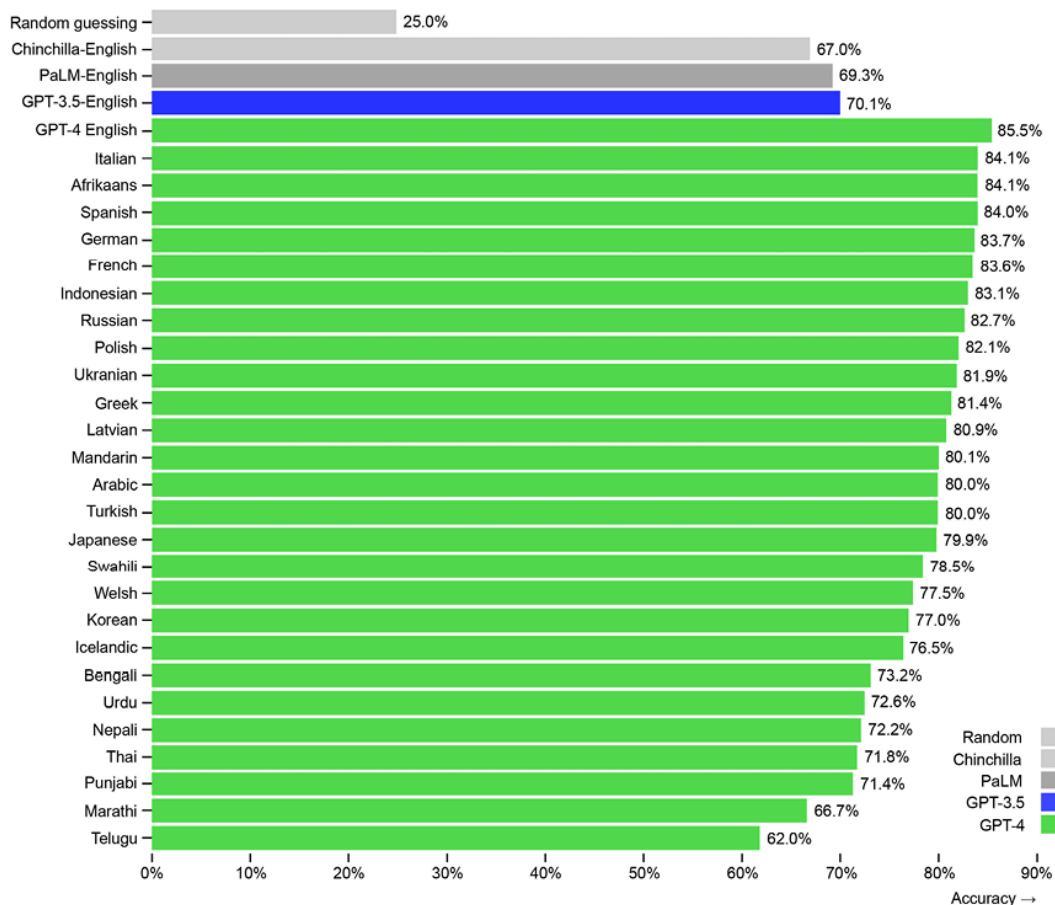
GPT-4 3-shot accuracy on MMLU across languages

Figure 3.2: GPT-4 3-shot accuracy on MMLU across languages (source: <https://openai.com/research/gpt-4>)

In addition to MMLU, GPT-4 has been benchmarked on a variety of SOTA systems and academic exams, as you can see from the following graph:

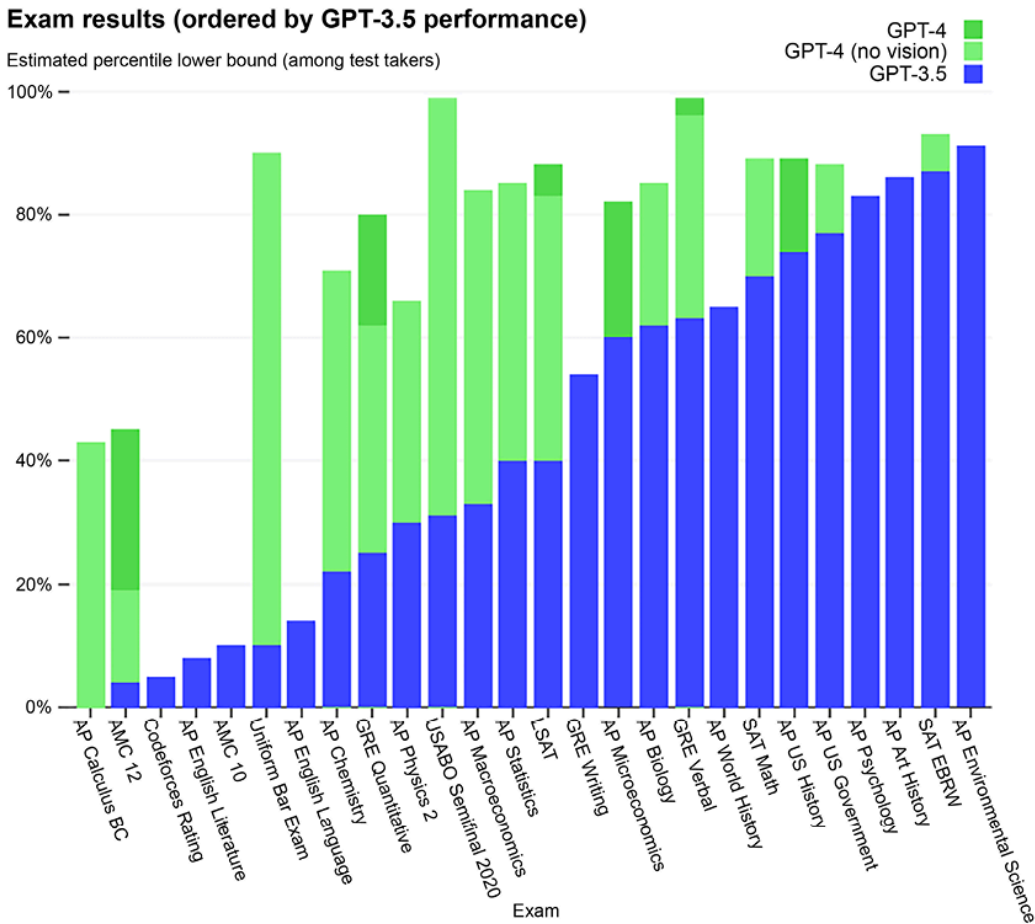


Figure 3.3: GPT performance on academic and professional exams (source: <https://arxiv.org/pdf/2303.08774.pdf>)



Note: in the preceding graph, you can see two versions of GPT-4, vision and no vision (along with the GPT-3.5 for benchmarking purposes). This is because GPT-4 is a multi-modal model, meaning that it can take images as input, in addition to text. However, in this chapter, we will benchmark only its textual capabilities.

Another great improvement of GPT-4 with respect to its predecessors (GPT-3.5 and GPT-3) is its noticeable reduction in the risk of hallucination.

Definition



Hallucination is a term that describes a phenomenon where LLMs generate text that is incorrect, nonsensical, or not real, but appears to be plausible or coherent. For example, an LLM may hallucinate a fact that contradicts the source or common knowledge, a name that does not exist, or a sentence that does not make sense.

Hallucination can happen because LLMs are not databases or search engines that store or retrieve factual information. Rather, they are statistical models that learn from massive amounts of text data and produce outputs based on the patterns and probabilities they have learned. However, these patterns and probabilities may not reflect the truth or the reality, as the data may be incomplete, noisy, or biased. Moreover, LLMs have limited contextual understanding and memory, as they can only process a certain number of tokens at a time and abstract them into latent representations. Therefore, LLMs may generate text that is not supported by any data or logic but is the most likely or correlated from the prompt.

In fact, even though it is still not 100% reliable, GPT-4 made great improvements with TruthfulQA benchmarks, which test the model's ability to separate fact from incorrect statements (we covered TruthfulQA benchmarks in *Chapter 1*, in the *Model evaluation* section).

Here, you can see an illustration that compares GPT-4 results in a TruthfulQA benchmark with those of GPT-3.5 (the model behind OpenAI's ChatGPT) and Anthropic-LM (we will cover this latter model in the next sections).

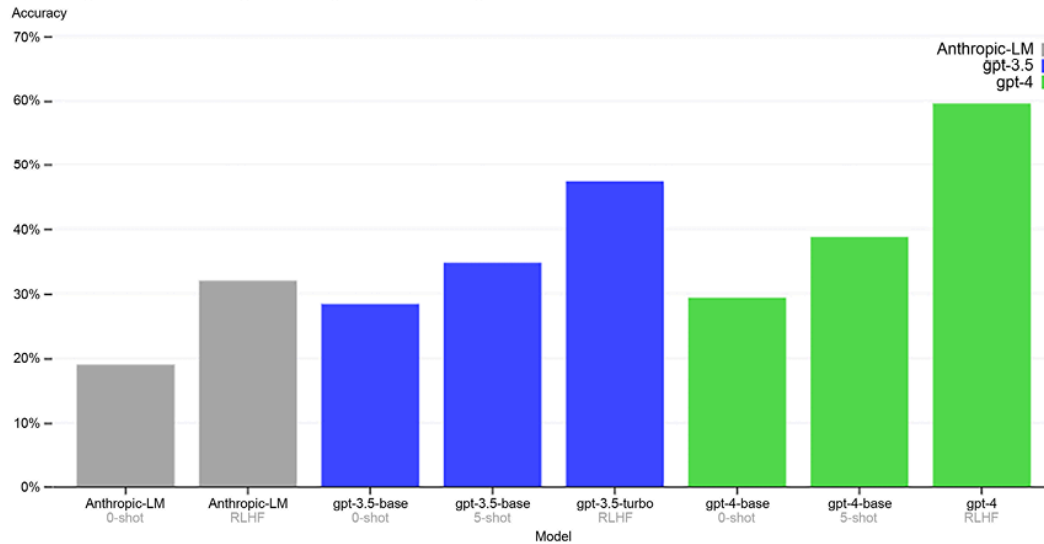
Accuracy on adversarial questions (TruthfulQA mc1)

Figure 3.4: Model comparison in TruthfulQA benchmark (source: <https://openai.com/research/gpt-4>)

Finally, with GPT-4, OpenAI made an additional effort to make it safer and more aligned, engaging from the beginning a team of over 50 experts in domains like AI alignment risks, privacy, and cybersecurity, with the goal of understanding the extent of the risks of such a powerful model and how to prevent them.

Definition



Alignment is a term that describes the degree to which LLMs behave in ways that are useful and harmless for their human users. For example, an LLM may be aligned if it generates text that is accurate, relevant, coherent, and respectful. An LLM may be misaligned if it generates text that is false, misleading, harmful, or offensive.

Thanks to this analysis, further data have been collected and used while training GPT-4 to mitigate its potential risks, resulting in a reduced risk compared to its predecessor, GPT-3.5.

Gemini 1.5

Gemini 1.5 is a SOTA generative AI model developed by Google and released in December 2023. Like GPT-4, Gemini is designed to be multi-modal, meaning that it can process and generate content across various modalities, including text, images, audio, video, and code. It is based on a **mixture-of-expert (MoE)** transformer.

Definition



In the context of transformer architecture, MoE refers to a model that incorporates multiple specialized sub-models, known as “experts,” within its layers. Each expert is a neural network designed to handle different types of data or tasks more efficiently. The MoE model uses a gating mechanism or router to determine which expert should process a given input, allowing the model to dynamically allocate resources and specialize in processing certain types of information. This approach can lead to more efficient training and inference, as it enables the model to scale up in size and complexity without a proportional increase in computational cost.

Gemini comes in various sizes, including Ultra, Pro, and Nano, to cater to different computational needs, from data centers to mobile devices. To use Gemini, developers can access it via the APIs provided for different model variants, allowing the integration of its capabilities into applications.

Compared to its previous version, Gemini 1.0, the current model outperforms it in text, vision, and audio tasks, as shown in the following screenshot:

Core Capability		Relative to	
		1.0 Pro	1.0 Ultra
Text	Math, Science & Reasoning	+28.9%	+5.2%
	Multilinguality	+22.3%	+6.7%
	Coding	+8.9%	+0.2%
	Instruction following	+9.2%	+2.5%
Vision	Image understanding	+6.5%	-4.1%
	Video understanding	+16.9%	+3.8%
Audio	Speech recognition	+1.2%	-5.0%
	Speech translation	+0.3%	-2.2%

Figure 3.5: Gemini 1.5 Pro and Ultra compared to its previous version 1.0 (source: https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf)

Similarly, it has demonstrated outstanding capabilities in domains such as math, science, and reasoning, and coding and multilinguality:

Capability	Benchmark	Gemini		
		1.0 Pro	1.0 Ultra	1.5 Pro
Math, Science & Reasoning	Hellaswag (Zellers et al., 2019)	84.7% 10-shot	87.8% 10-shot	92.5% 10-shot
	MMLU: Multiple-choice questions in 57 subjects (professional & academic). (Hendrycks et al., 2021a)	71.8% 5-shot	83.7% 5-shot	81.9% 5-shot
	GSM8K: Grade-school math problems. (Cobbe et al., 2021)	77.9% 11-shot	88.9% 11-shot	91.7% 11-shot
	MATH: Math problems ranging across 5 levels of difficulty and 7 sub-disciplines. (Hendrycks et al., 2021b)	32.6% 4-shot Minerva prompt	53.2% 4-shot Minerva prompt	58.5% 4-shot Minerva prompt 59.4% 7-shot
	AMC 2022-23: 250 latest problems including 100 AMC 12, 100 AMC 10, and 50 AMC 8 problems.	22.8% 4-shot	30% 4-shot	37.2% 4-shot
	BigBench - Hard: A subset of harder tasks from Big Bench formatted as CoT problems. (Srivastava et al., 2022)	75.0% 3-shot	83.6% 3-shot	84.0% 3-shot
	DROP: Reading comprehension & arithmetic. (Metric: F1-Score). (Dua et al., 2019)	74.1% Variable shots	82.4% Variable shots	78.9% Variable shots
Coding	HumanEval chat preamble* (Metric: pass rate). (Chen et al., 2021)	67.7% 0-shot (PT)	74.4% 0-shot (PT)	71.9% 0-shot
	Natural2Code chat preamble* (Metric: pass rate).	69.6% 0-shot	74.9% 0-shot	77.7% 0-shot
Multilinguality	WMT23: sentence-level machine translation (Metric: BLEURT). (Tom et al., 2023)	71.73 (PT) 1-shot	74.41 (PT) 1-shot	75.20 1-shot
	MGSM: multilingual math reasoning. (Shi et al., 2023b)	63.45% 8-shot (PT)	78.95% 8-shot (PT)	88.73% 8-shot

Figure 3.6: Gemini 1.5 Pro compared to Gemini 1.0 Pro and Ultra on different benchmarks (source: https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf)

Note that Gemini 1.5 Pro is outperforming Gemini 1.0 Ultra (which is remarkably bigger) in many benchmarks across the various domains. As of today, Gemini Pro can be tried via a web app at gemini.google.com for free, while Gemini Ultra is available via a premium subscription with a monthly fee. On the other hand, Gemini Nano, which is tailored for mobile devices, can be executed on capable Android devices

via the Google AI Edge SDK for Android. Note that, as of April 2024, this SDK is still under early access preview and you can apply for the early access program at

https://docs.google.com/forms/d/e/1FAIpQLSdDvg0eEzcUY_-CmtiMZLd68KD3F0usCnRzKKzWb4sAYwhFJg/viewform. Finally,

Gemini Pro and Ultra can also be consumed by developers via the REST API from Google AI Studio.

Claude 2

Claude 2, which stands for Constitutional Large-scale Alignment via User Data and Expertise, is an LLM developed by Anthropic, a research company founded by former OpenAI researchers and focused on AI safety and alignment. It was announced in July 2023.

Claude 2 is a transformer-based LLM that has been trained on a mix of publicly available information from the internet and proprietary data, via unsupervised learning, RLHF, and **constitutional AI (CAI)**.

CAI is a real peculiarity of Claude. In fact, Anthropic paid extraordinary attention to Claude 2 alignment with safety principles. More specifically, Anthropic developed this unique technique called CAI, which was disclosed in December 2022 in the paper *Constitutional AI: Harmlessness from AI Feedback*.

CAI aims to make the model safer and more aligned with human values and intentions by preventing toxic or discriminatory output, not helping a human engage in illegal or unethical activities, and broadly creating an AI system that is helpful, honest, and harmless. To achieve this, it uses a set of principles to guide the model's behavior and outputs, rather than relying on human feedback or data alone. The principles are derived from various sources, such as the UN Declaration of Human Rights, trust and safety best practices, principles proposed by other AI research labs, non-Western perspectives, and empirical research.

CAI uses these principles in two stages of the training process:

- First, the model is trained to critique and revise its own responses using the principles and a few examples.
- Second, the model is trained via reinforcement learning, but rather than using human feedback, it uses AI-generated feedback based on the principles to choose the more harmless output.

The following illustration shows the training process according to the CAI technique:

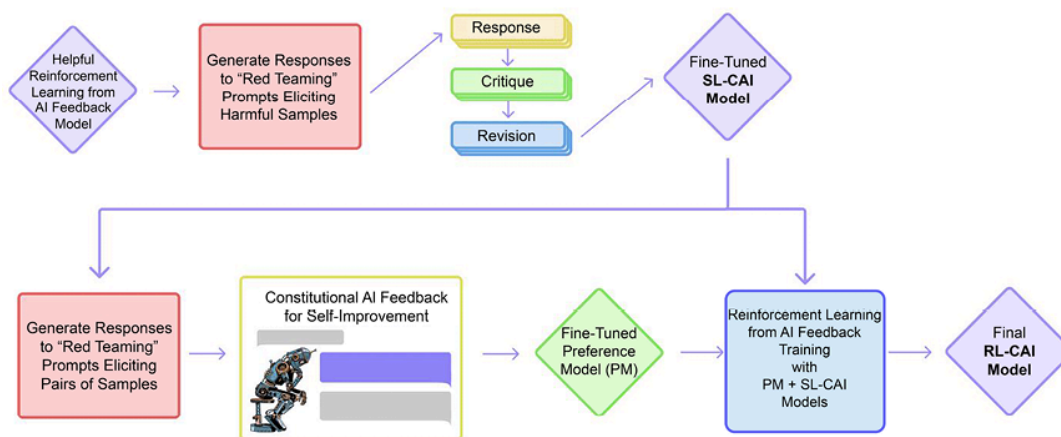


Figure 3.7: Claude's training process according to the CAI technique
(source: <https://arxiv.org/abs/2212.08073>)

Another peculiarity of Claude 2 is the context length, which has a limit of 100,000 tokens. This means that users can input longer prompts, namely pages of technical documentation or even a book, which do not need to be embedded. Plus, the model can also generate longer output compared to other LLMs.

Finally, Claude 2 demonstrates relevant capabilities also when working with code, scoring 71.2% on the HumanEval benchmark.

Definition



HumanEval is a benchmark for evaluating the code generation ability of LLMs. It consists of 164 human-crafted coding problems in Python, each with a prompt, a solution, and a test suite. The problems cover various topics, such as data structures, algorithms, logic, math, and string manipulation. The benchmark can be used to measure the functional correctness, syntactic validity, and semantic coherence of the LLM’s outputs.

Overall, Claude 2 is a very interesting model and competitor of GPT-4 to pay attention to. It can be consumed via the REST API or directly via the Anthropic beta chat experience (limited for US and UK users as of August 2023).

The following comparison table shows the main differences between the three models:

	GPT-4	Gemini	Claude 2
Company or institution	OpenAI	Google	Anthropic
First release	March 2023	December 2023	July 2023
Architecture	Transformer-based, de-coder only	Transformer-based	Transformer-based

Sizes and variants	Parameters not officially specified Two context-length variants: GPT-4 8K tokens GPT-4 32K tokens	Three sizes, from smallest to largest: Nano, Pro, and Ultra	Not officially specified
How to use	REST API at OpenAI developer platforms Using OpenAI Playground at https://platform.openai.com/playground	REST API at Google AI Studio Using Gemini at https://gemini.google.com/	REST API after compiling the form at https://www.anthropic.com/claude

Table 3.1: Comparison table of GPT-4, PaLM 2, and Claude 2

In addition to proprietary models, there is a huge market for open-source LLMs available today. Let’s discuss some of these in the next section.

Open-source models

The advantage of an open-source model is that, by definition, developers have full visibility and access to the source code. In the context of LLMs, this implies the following:

- You have major control over the architecture, meaning that you can also modify it in the local version you are going to use within your project. This also implies that they are not prone to potential updates to the source code made by models' owners.
- There is the possibility to train your model from scratch, on top of the classical fine-tuning, which is also available for proprietary models.
- Free to use, meaning that you won't incur any charge while using those LLMs, in contrast with the proprietary ones that have pay-per-use pricing.

To compare open-source models, throughout this book, we will refer to the independent Hugging Face Open LLM Leaderboard (you can find it at

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard), a project that aims to evaluate and compare the performance of LLMs on various **natural language understanding** (NLU) tasks. The project is hosted on Hugging Face Spaces, a platform for creating and sharing machine-learning applications.

The Open LLM Leaderboard uses four main evaluation benchmarks, which we covered in *Chapter 1*, in the *Model evaluation* section:

- **AI2 Reasoning Challenge (ARC)**: Grade-school science questions and complex NLU tasks.
- **HellaSwag**: Common sense reasoning.
- **MMLU**: Tasks in various domains, including math, computer science, and law.
- **TruthfulQA**: An evaluation of how truthful the model is when generating answers.

Even though those are just a subsample of the plethora of LLMs' benchmarks, we will stick to this leaderboard as a reference evaluation framework as it being widely adopted.

LLaMA-2

Large Language Model Meta AI 2 (LLaMA-2) is a new family of models developed by Meta and unveiled to the public on July 18, 2023, open source and for free (its first version was originally limited to researchers).

It is an **autoregressive** model with an optimized, decoder-only transformer architecture.

Definition



The concept of autoregressive in the context of transformers refers to the fact that the model predicts the next token in the sequence, conditioned on all the previous tokens. This is done by masking the future tokens in the input so that the model can only attend to the past tokens. For example, if the input sequence is “The sky is blue,” the model would predict “The” first, then “sky,” then “is,” and finally “blue,” using a mask to hide the tokens that come after each prediction.

LLaMA-2 models come in three sizes: 7, 13, and 70 billion parameters. All the versions have been trained on 2 trillion tokens and have a context length of 4,092 tokens.

On top of that, all model sizes come with a “chat” version, called LLaMA-2-chat, which is more versatile for general-purpose conversational scenarios compared to the base model LLaMA-2.

Note

In the context of LLMs, the difference between **base models** and “chat” or **assistant models** is primarily in their training and intended use:



- **Base models:** These models are trained on vast amounts of text data, often sourced from the internet, and their primary function is to predict the next word in a given context, which makes them great at understanding and generating language. However, they might not always be precise or focused on specific instructions.
- **Assistant models:** These models start as base LLMs but are further fine-tuned with input-output pairs that include instructions and the model's attempts to follow those instructions. They often employ RLHF to refine the model, making it better at being helpful, honest, and harmless. As a result, they are less likely to generate problematic text and are more suitable for practical applications like chatbots and content generation. For example, the assistant model GPT-3.5 Turbo (the model behind ChatGPT) is a fine-tuned version of the completion model GPT-3.

In essence, while base models provide a broad understanding of language, assistant models are optimized to follow instructions and provide more accurate and contextually relevant responses.

LLaMA-2-chat was developed with a fine-tuning process that consisted of two main steps:

1. **Supervised fine-tuning:** This step involves fine-tuning the model on publicly available instruction datasets and over 1 million human annotations, to make them more helpful and safe for conversational use cases. The fine-tuning process uses a selected list of prompts to guide the model outputs, and a loss function that encourages diversity and relevance (that's the reason why it is "supervised").

2. **RLHF**: As we saw while introducing GPT-4, RLHF is a technique that aims at using human feedback as an evaluating metric for LLMs' generated output, and then using that feedback to further optimize the model.

The following is an illustration of how the training process for LLaMA works:

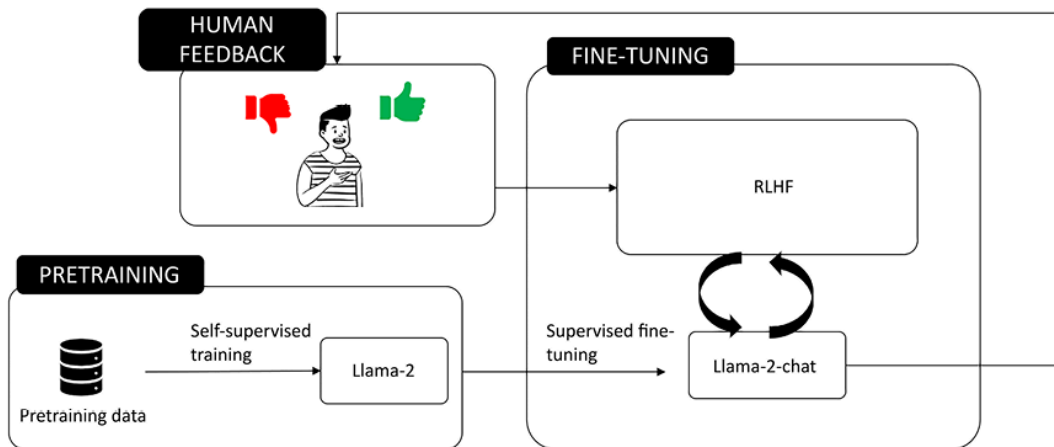


Figure 3.8: Two-step fine-tuning to obtain LLaMa-2 chat (source: <https://ai.meta.com/resources/models-and-libraries/llama/>)

To access the model, you need to submit a request on Meta's website (the form is available at <https://ai.meta.com/resources/models-and-libraries/llama-downloads/>). Once a request is submitted, you will receive an email with the GitHub repository where you will be able to download the following assets:

- Model code
- Model weights
- README (User Guide)
- Responsible Use Guide
- License
- Acceptable Use Policy
- Model Card

Falcon LLM

Falcon LLM is a representation of a new trend of LLMs, consisting of building lighter models (with fewer parameters) and focusing rather on the quality of the training dataset. Indeed, it is a matter of fact that complex models like GPT-4 with trillions of parameters are extremely heavy, both in the training phase and inference phase. This implies the need for high and expensive computational power (GPU and TPU-powered) as well as a long training time.

Falcon LLM is an open-source model launched by Abu Dhabi's **Technology Innovation Institute (TII)** in May 2023. It is an autoregressive, decoder-only transformer, trained on 1 trillion tokens, and it has 40 billion parameters (even though it has also been released as a lighter version with 7 billion parameters). Similarly to what we saw for LLaMA, Falcon LLM also comes with a fine-tuned variant, called "Instruct," which is tailored toward following the user's instructions.

Definition



Instruct models are specialized for short-form instruction following. Instruction following is a task where the model has to execute a natural language command or query, such as "write a haiku about cats" or "tell me about the weather in Paris." The Instruct fine-tuned models are trained on a large dataset of instructions and their corresponding outputs, such as the Stanford Alpaca dataset.

According to the Open LLM leaderboard, since its launch, Falcon LLM has been among the first positions globally, second only to some versions of LLaMA.

So, the question might be: how can a model with "only" 40 billion parameters perform so well? In fact, the answer is in the quality of the dataset. Falcon was developed using specialized tools and incorporates

a unique data pipeline, which is capable of extracting valuable content from web data. The pipeline was designed to extract high-quality content by employing extensive filtering and deduplication techniques. The resulting dataset, called *RefinedWeb*, has been released by TII under the Apache-2.0 license and can be found at <https://huggingface.co/datasets/tiiuae/falcon-refinedweb>.

By combining superior data quality with these optimizations, Falcon achieves remarkable performance while utilizing around 75% and 80% of the training compute budget of GPT-3 and PaLM-62B, respectively.

Mistral

The third and last open-source model series we are going to cover is Mistral, developed by Mistral AI, a company founded in April 2023 by a team of AI scientists who previously worked at Meta Platforms and Google DeepMind. Based in France, the company has quickly made a name for itself by raising significant funding and releasing open-source LLMs, emphasizing the importance of transparency and accessibility in AI development.

The Mistral model, particularly the Mistral-7B-v0.1, is a decoder-only transformer with 7.3 billion parameters, designed for generative text tasks. It’s known for its innovative architecture choices like **grouped-query attention (GQA)** and **sliding-window attention (SWA)**, which have allowed it to outperform other models in benchmarks.

Definition	
	GQA and SWA are mechanisms designed to improve the efficiency and performance of an LLM.
	GQA is a technique that allows for faster inference times compared to standard full attention mechanisms. It does this by partitioning the attention mechanism’s query



heads into groups, with each group sharing a single key head and value head.

SWA is used to handle longer text sequences efficiently. It extends the model's attention beyond a fixed window size, allowing each layer to reference a range of positions from the preceding layer. This means that the hidden state at a certain position in one layer can attend to hidden states within a specific range in the previous layer, thus enabling the model to access tokens at a greater distance and manage sequences of varying lengths with a reduced inference cost.

The model also provides a variant that was fine-tuned for general-purpose capabilities. This variant is called Mistral-7B-instruct, which outperformed all other 7 billion LLMs on the market (as of April 2024) on MT-Bench (an evaluation framework that uses an LLM as a judge).

Like many other open-source models, Mistral can be consumed and downloaded via Hugging Face Hub.

Note



In February 2024, Mistral AI and Microsoft entered a multi-year partnership to accelerate AI innovation. This collaboration will leverage Microsoft's Azure AI supercomputing infrastructure to support the development and deployment of Mistral AI's LLMs. Mistral AI's models, including their advanced model, Mistral Large, will be available to customers through Azure AI Studio and Azure Machine Learning model catalog. The partnership aims to expand Mistral AI's reach to global markets and foster ongoing research collaboration.

The following comparison table provides the main differences between the three models:

	LlaMA	Falcon LLM	Mistral
Company or institution	Meta	Technology Innovation Institute (TII)	Mistral AI
First release	July 2023	May 2023	September 2023
Architecture	Autoregressive transformer, decoder-only	Autoregressive transformer, decoder-only	Transformer, decoder only
Sizes and variants	Three sizes: 7B, 13B, and 70B, alongside the fine-tuned version (chat)	Two sizes: 7B and 40B, alongside the fine-tuned version (instruct)	7B size alongside the fine-tuned version (instruct)
Licenses	A custom commercial license is available at https://ai.meta.com/resources/models-and-libraries/llama-downloads/	Commercial Apache 2.0 licensed	Commercial Apache 2.0 licensed

How to use	Submit re- quest form at https://ai. meta.com/re sources/mod els-and-li- braries/lla ma-down- loads/ and download the GitHub repo Also available in Hugging Face Hub	Download or use Hugging Face Hub Inference API/Endpoint	Download or use Hugging Face Hub Inference API/Endpoint or Azure AI Studio
------------	---	---	--

Table 3.2: Comparison table of LLMs

Beyond language models

So far, we have only been covering language-specific foundation models as they are the focus of this book. Nevertheless, in the context of AI-powered applications, it is worth mentioning that there are additional foundation models that can handle data that is different from text, which can be embedded and orchestrated.

Here, you can find some examples of **large foundation models (LFMs)** on the market today:

- **Whisper:** It is a general-purpose speech recognition model developed by OpenAI that can transcribe and translate speech in multiple languages. It is trained on a large dataset of diverse audio and is also a multitasking model that can perform multilingual speech recognition, speech translation, spoken language identification, and voice activity detection.

- **Midjourney:** Developed by the independent research lab of the same name, Midjourney is based on a sequence-to-sequence transformer model that takes text prompts and outputs a set of four images that match the prompts. Midjourney is designed to be a tool for artists and creative professionals, who can use it for rapid prototyping of artistic concepts, inspiration, or experimentation.
- **DALL-E:** Similar to the previous one, DALL-E, developed by OpenAI, generates images from natural language descriptions, using a 12-billion parameter version of GPT-3 trained on a dataset of text-image pairs.

The idea is that we can combine and orchestrate multiple LFM's within our applications to achieve extraordinary results. For example, let's say we want to write a review about an interview with a young chef and post it on Instagram. The involved models might be the following:

- **Whisper** will convert the interview audio into a transcript.
- An **LLM**, such as Falcon-7B-instruct, with a web plugin, will extrapolate the name of the young chef and search it on the internet to retrieve the biography.
- Another **LLM**, such as LLaMA, will process the transcript and generate a review with an Instagram post style. We can also ask the same model to generate a prompt that will ask the following model to generate a picture based on the post content.
- **Dall-E** will generate an image based on the prompt generated by the LLM.

We will then provide our LFM's flow with an Instagram plugin so that the application is able to post the whole review, including the illustration, on our profile.

Finally, there are emerging LFM's that are meant to be multi-modal, meaning that they can handle multiple data formats with just one architecture. An example is GPT-4 itself.

The following screenshot shows an example of an early OpenAI experiment with GPT-4 visuals, demonstrating its understanding of funny aspects within an image:

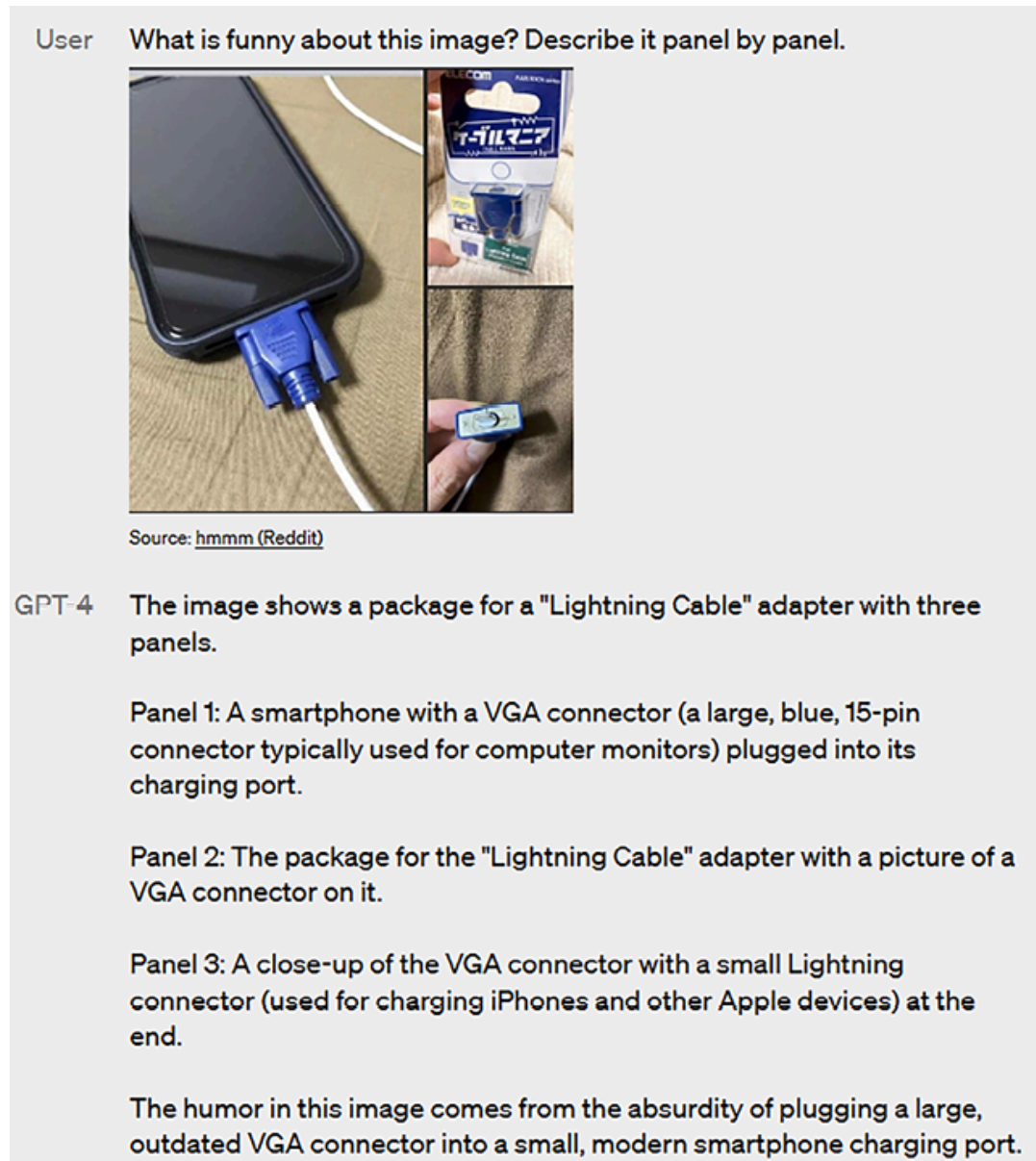
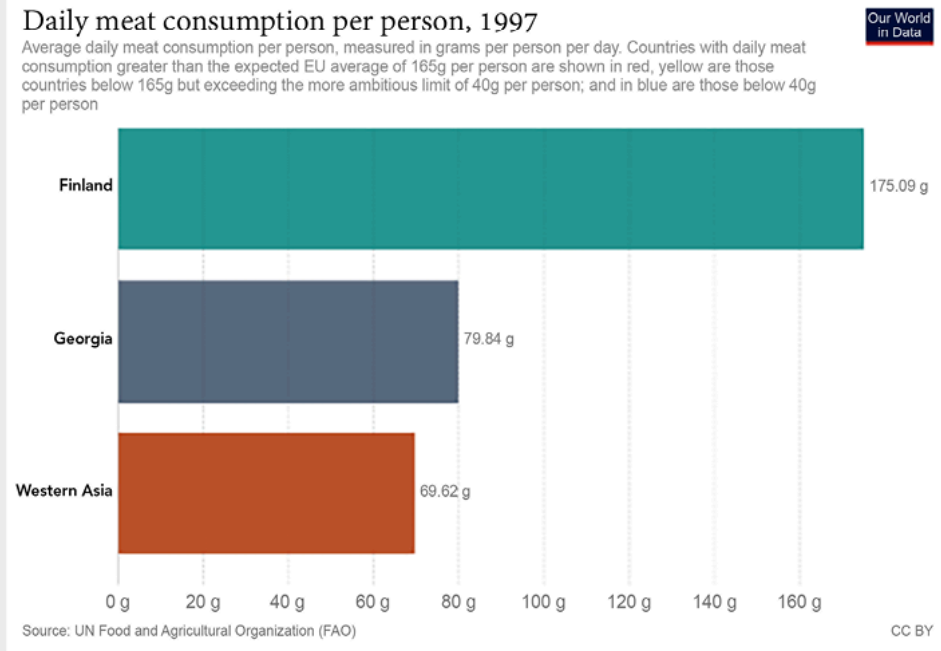


Figure 3.9: Early experiments with GPT-4 visuals (source: <https://openai.com/research/gpt-4>)

The following screenshot shows another example of an earlier version of GPT-4, demonstrating how it could understand and explain graphs in detail:

User What is the sum of average daily meat consumption for Georgia and Western Asia? Provide a step-by-step reasoning before providing your answer.



GPT-4 To find the sum of average daily meat consumption for Georgia and Western Asia, we will follow these steps:

1. Identify the average daily meat consumption for Georgia.
2. Identify the average daily meat consumption for Western Asia.
3. Add the values from steps 1 and 2.

Step 1: From the provided chart, we see that the average daily meat consumption for Georgia is 79.84 grams per person per day.

Step 2: From the chart, we see that the average daily meat consumption for Western Asia is 69.62 grams per person per day.

Step 3: Add the values from steps 1 and 2.

$$79.84 \text{ g (Georgia)} + 69.62 \text{ g (Western Asia)} = 149.46 \text{ g}$$

So, the sum of average daily meat consumption for Georgia and Western Asia is 149.46 grams per person per day.

Figure 3.10: Early experiments with GPT-4 visuals (source: <https://openai.com/research/gpt-4>)

The following example shows how an early version of GPT-4 could understand and solve complex mathematical problems while also providing the corresponding justification for its response:

User Answer question I.1.a. Think step-by-step.

I. Principe de la détection de rayonnement avec un bolomètre

Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température T , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance $R(T)$ varie avec la température T ; cette variation est caractérisée par le coefficient $\alpha = \frac{1}{R} \frac{dR}{dT}$. L'ensemble possède la capacité thermique C_{th} .

Un barreau, conducteur thermique, homogène, de longueur L , de section S et de conductivité thermique λ et sans échanges thermiques latéraux, relie le bolomètre à un thermostat de température T_b fixe.

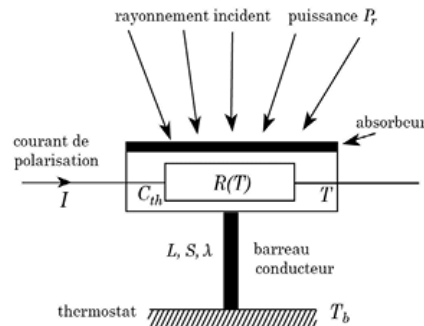


Figure 1 - Schéma d'un bolomètre

I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement $T = T_0$.

I.1.a) Exprimer la température $T(x)$ en tout point du barreau conducteur repéré par son abscisse x ; on prendra $T(0) = T_0$ et $T(L) = T_b$.

GPT-4 To answer question I.1.a, we need to express the temperature $T(x)$ in every point of the conductive bar, which is characterized by the abscissa x . We are given that $T(0) = T_0$ and $T(L) = T_b$.

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:

$$d^2T/dx^2 = 0$$

Integrating once, we get:

$$dT/dx = C_1$$

Integrating a second time:

$$T(x) = C_1 x + C_2$$

To determine the constants C_1 and C_2 , we use the boundary conditions:

$$T(0) = T_0, \text{ which gives } C_2 = T_0$$

$$T(L) = T_b, \text{ which gives } C_1 = (T_b - T_0) / L$$

Finally, the expression for $T(x)$ is:

$$T(x) = (T_b - T_0) * (x / L) + T_0$$

This represents a linear temperature profile along the conductive bar, with a slope of $(T_b - T_0) / L$.

Figure 3.11: Early experiments with GPT-4 visuals (source: <https://openai.com/research/gpt-4>)

GPT-4 is just one example of a **large multimodal model (LMM)**, and it is representative of the trend that we will probably witness in the next

few years.

A decision framework to pick the right LLM

In previous paragraphs, we covered some of the most promising LLMs available in the market today. Now, the question is: which one should I use within my applications? The truth is that there is not a straightforward answer to this question.

Considerations

There are many factors to consider when choosing an LLM for your application. Those factors also need to be declined in two scenarios: proprietary and open-source LLMs. The following are some factors and trade-offs you might want to consider while choosing your LLMs:

- **Size and performance:** We saw that more complex models (that means, with a high number of parameters) tend to have better performance, especially in terms of parametric knowledge and generalization capabilities. Nevertheless, the larger the model, the more computation and memory it requires to process the input and generate the output, which can result in higher latency and, as we will see, higher costs.
- **Cost and hosting strategy:** When incorporating LLMs within our applications, there are two types of costs we have to keep in mind:
 - **Cost for model consumption:** This refers to the fee we pay to consume the model. Proprietary models like GPT-4 or Claude 2 require a fee, which is typically proportional to the number of tokens processed. On the other hand, open-source models like LLaMA or Falcon LLM are free to use.
 - **Cost for model hosting:** This refers to your hosting strategy. Typically, proprietary models are hosted in a private or public hyperscaler, so that they can be consumed via a REST API and you don't have to worry about the underlying infrastructure

(for example, GPT-4 is hosted in a super-computer built in the Microsoft Azure cloud). With open-source models, we typically need to provide our own infrastructure, since those models can be downloaded locally. Of course, the larger the model, the more powerful the computational power needed.

Note

In the context of open-source models, another option to consume those models is that of using the Hugging Face Inference API. The free version allows you to test and evaluate, with a limited rate, all the available LLMs on a shared infrastructure hosted on Hugging Face. For production use cases, Hugging Face also offers Inference Endpoints, so that you can easily deploy your LLMs on a dedicated and fully managed infrastructure, with the possibility to configure parameters like region, compute power, and security level to accommodate your constraints in terms of latency, throughput, and compliance.

Pricing for the Inference Endpoint is publicly available at <https://huggingface.co/docs/inference-endpoints/pricing>.

- **Customization:** This might be a requirement you want to evaluate before deciding which model to adopt. In fact, not all models are equally flexible in terms of customization. When we talk about customization, we refer to two activities:
 - **Fine-tuning:** This is the process of slightly adjusting LLMs' parameters to better fit into a domain. All open-source models can be fine-tuned. When it comes to proprietary models, not all LLMs can be fine-tuned: for example, OpenAI's GPT-3.5 can be fine-tuned, while the process of fine-tuning the GPT-4-0613 is still experimental and accessible under request to OpenAI (as per December 2023).

Henceforth, it is important to understand whether you will need fine-tuning in your application and decide accordingly.

- **Training from scratch:** If you really want an LLM that is super specific about your domain knowledge, you might want to re-train the model from scratch. To train an LLM from scratch, without having to reinvent an architecture, you can download open-source LLMs and simply re-train them on custom datasets. Of course, this implies that we have access to the source code, which is not the case when we work with proprietary LLMs.
- **Domain-specific capabilities:** We saw that the most popular way of evaluating LLMs' performance is that of averaging different benchmarks across domains. However, there are benchmarks that are tailored towards specific capabilities: if MMLU measures LLMs' generalized culture and commonsense reasoning, TruthfulQA is more concerned with LLMs' alignment, while HumanEval is tailored towards LLMs' coding capabilities.

Henceforth, if you have a tailored use case in mind, you might want to use a model that is a top performer in one specific benchmark, rather than a top performer, on average, across all benchmarks. Namely, you might pick Claude 2 if you are looking for exceptional coding capabilities, or PaLM 2 if analytical reasoning is what you are looking for. On the other hand, if you need a model that encompasses all of these capabilities, GPT-4 might be the right choice for you.

Picking a domain-specific model is also a way to make some savings in terms of model complexity. The thing is, it might be sufficient for you to use a relatively small model (for example, a LLaMA-7B-instruct) if you need to use it for a specific use case, which comes with all the benefits in terms of cost and performance.

Note

If you are looking for LLMs that are *extremely* specific, there is a plethora of models that have been trained on domain-specific technical documentation. For example, at the beginning of 2023, the **Stanford Center for Research**



on Foundation Models (CRFM) and MosaicML announced the release of BioMedLM, a decoder-only transformer-based LLM with 2.7 billion parameters, trained on biomedical abstracts and papers.

Another example is BloombergGPT, a 50 billion parameter LLM specialized for the financial domain developed by Bloomberg and trained on a 363 billion token dataset based on Bloomberg's extensive data sources, perhaps the largest domain-specific dataset yet, augmented with 345 billion tokens from general purpose datasets.

To make this decision framework more practical, let's consider the following imaginary case study about the company TechGen.

Case study

TechGen Solutions, a leading provider of AI-driven analytics, face a decision between two advanced language models for their next-generation customer interaction system: GPT-4 and LLaMa-2. They require a robust language model that can handle diverse customer queries, provide accurate technical information, and integrate with their proprietary software. The following are their options:

- **GPT-4:** Developed by OpenAI, GPT-4 is known for its vast parameter count and the ability to process both text and image inputs
- **LLama 2:** Created by Meta AI, LLama 2 is an open-source model praised for its accessibility and performance on a smaller dataset.

The following are the factors that they consider when making their decision:

- **Performance:** TechGen evaluates the models' performance, particularly in generating technical content and code, where GPT-4 has shown higher accuracy.

- **Integration:** The ease of integration with TechGen's systems is critical, with GPT-4 potentially offering more seamless compatibility due to its widespread adoption.
- **Cost:** While LLama 2 is free for commercial use under certain conditions, GPT-4 comes with a cost, which TechGen must factor into their decision.
- **Future-proofing:** TechGen considers the long-term viability of each model, including the potential for updates and improvements.

Based on these considerations, TechGen opts for GPT-4, swayed by its superior performance in generating complex, technical responses and its multilingual capabilities, which align with their international expansion plans. The decision is also influenced by GPT-4's image processing feature, which TechGen anticipates will become increasingly relevant as they incorporate more multimedia content into their customer service.

TechGen's choice of GPT-4 over LLama 2 is driven by the need for a high-performing, versatile language model that can scale with their growing global presence and diverse customer needs. While LLama 2's open-source nature and cost effectiveness are appealing, GPT-4's advanced capabilities and future-proof features present a more compelling case for TechGen's ambitious goals.

Note that these decision factors are not meant to be an exhaustive guide to deciding which models to embed within applications. Nevertheless, those are useful elements of reflection while setting up your application flow, so that you can determine your requirements and then shortlist those LLMs that are more suitable for your goals.

Summary

This chapter covered some of the most promising LLMs in the market. It first differentiated between proprietary and open-source models, with all the related pros and cons. It then offered a deep dive into the architecture and technical features of GPT-4, PaLM-2, Claude 2,

LLaMa-2, Falcon LLM, and MPT, with the addition of a section covering some LMMs. Finally, it provided a light framework to help developers decide which LLMs to pick while building AI-powered applications. This is pivotal to get the greatest impact from your application, given your industry-specific scenario.

Starting from the next chapter, we will start working hands-on with LLMs within applications.

References

- GPT-4 Technical Report. <https://cdn.openai.com/papers/gpt-4.pdf>
- Train short, test long: attention with linear biases enables input length extrapolation. <https://arxiv.org/pdf/2108.12409.pdf>
- Constitutional AI: Harmlessness from AI Feedback. <https://arxiv.org/abs/2212.08073>
- Hugging Face Inference Endpoint. <https://huggingface.co/docs/inference-endpoints/index>
- Hugging Face Inference Endpoint Pricing. <https://huggingface.co/docs/inference-endpoints/pricing>
- Model Card for BioMedLM 2.7B. <https://huggingface.co/stanford-crfm/BioMedLM>
- PaLM 2 Technical Report. <https://ai.google/static/documents/palm2techreport.pdf>
- Solving Quantitative Reasoning Problems with Language Models. <https://arxiv.org/abs/2206.14858>
- Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. <https://arxiv.org/abs/2306.05685>

Join our community on Discord

Join our community's Discord space for discussions with the author and other readers:

<https://packt.link/llm>

