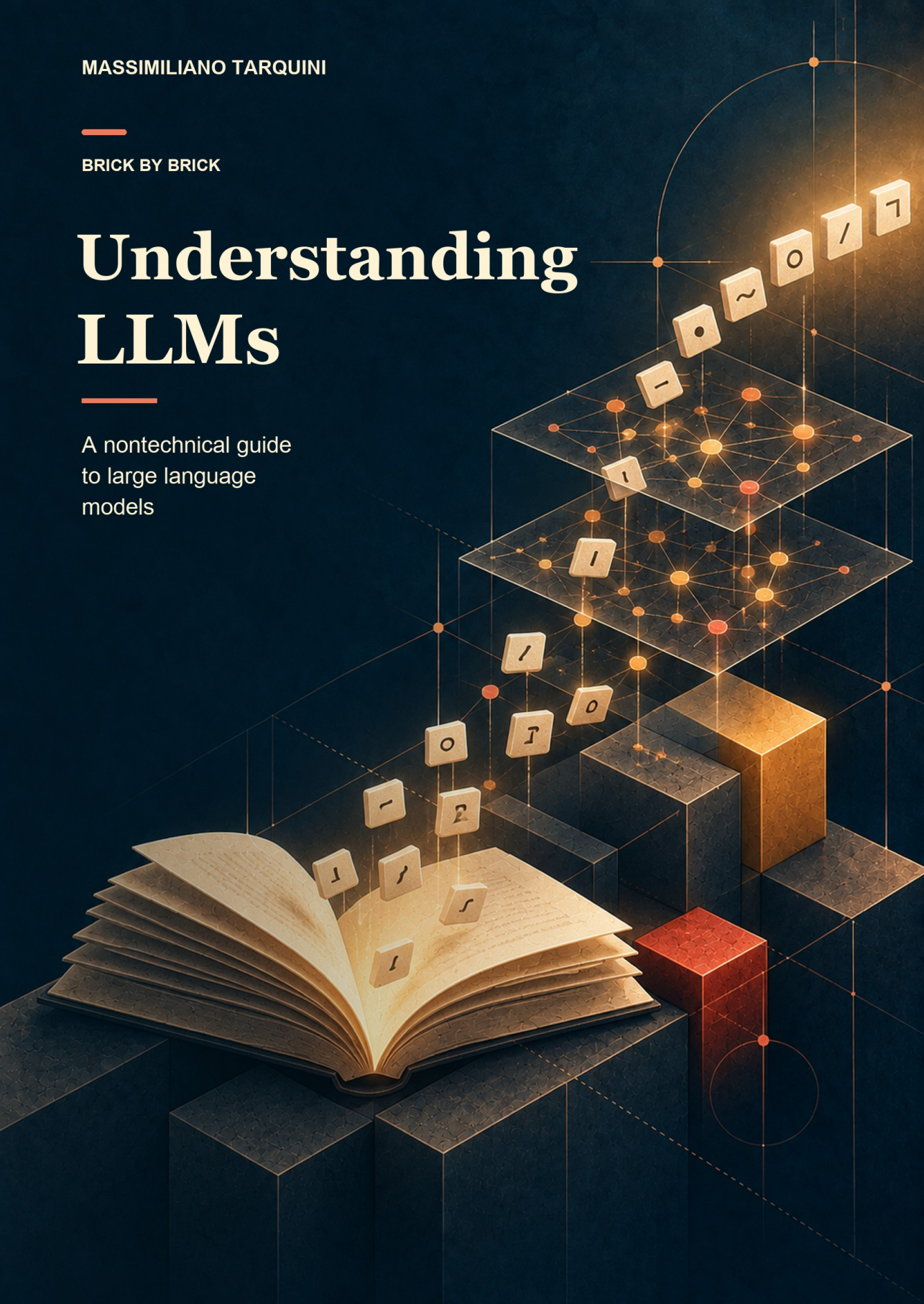
MASSIMILIANO TARQUINI

BRICK BY BRICK

Understanding LLMs

A nontechnical guide to large language models



BRICK BY BRICK

Understanding LLMs

A nontechnical guide

Massimiliano Tarquini

English edition · October 2026

Contents

Part I — Foundations 5

Introduction 5

What Are Large Language Models 5

How They Work Without Going into Too Much Detail 7

Part II — Capabilities and Interaction 9

What LLMs Can Do 9

Talking to LLMs — Why Prompts Matter and How Interfaces Are Designed 10

Emergent Abilities and How to Measure Them 12

Beyond Questions — How to Use Prompting to Harness Emergent Abilities 16

From Words to Abilities — Practical Examples of Emergent Prompting 18

Part III — Ethics, Limitations, and Society 20

Important Questions — What We Should Be Asking About LLMs 20

The Future of LLMs — Risks and Limitations of What They Cannot and Should Not Do 21

Using LLMs Wisely 21

Human-Centered AI and Explainability — Trusting What You Can Understand 24

Asking Models to Explain Themselves 25

Practical Examples of Explainability in Action 26

Myths and Reality — What LLMs Cannot Do Even When They Seem Able To 29

When Overconfidence Meets Artificial Intelligence 31

A Final Reflection 35

Part IV — Applications and Examples 36

Real-World Examples 36

Everyday Tools Quietly Powered by AI 37

Opportunities and Benefits — Why LLMs Can Be Truly Useful 37

Unexpected Stories from the World of Language Models 39

Contents

Part V — From General to Specialized Intelligence 41
From General to Specialized — How LLMs Learn New Skills 41
LLMs as Memory Machines — Retrieving Knowledge and Human Experience 42
A Whole World of Language Models — From Giants to Little Helpers 43
Beyond Words — Interpreting Images Alongside Text 45

Part VI — Infrastructure and Sustainability 47
Power and Price — The Hidden Cost of Intelligence 47
Comparing AI Models — Cloud Giants and Local Experts 48

Part VII — Appendices 50
Before You Ask — A Checklist for Using LLMs Wisely 50
Conclusion 50
Glossary 51
Want to Know More — Resources to Explore 52

Part I — Foundations

How to read this guide. This guide explains how to choose a use for a model, check its result, and recognize when another solution is needed. To follow the examples, you only need to read a short table and compare two texts; no programming knowledge is required. The journey includes explanations, verifiable references, and guided activities. Stories without corroboration are labeled as illustrative scenarios. This edition is under review.

Introduction

Artificial intelligence is no longer confined to science fiction: it is part of our everyday lives. You may already have interacted with it without realizing it, when your phone suggests words as you type, when a chatbot answers your questions, or when a tool helps you rewrite an email.

At the heart of many of these intelligent tools are Large Language Models, or LLMs for short. These artificial intelligence systems are changing the way we work, learn, and communicate. But what exactly are they? How do they work? And why are they attracting so much attention?

We will start with a simple question: how can a system produce a readable answer, and why can that answer be wrong? From there, we will work through the requests to make, the checks to perform, and the choices between tools. Technical terms will be explained when needed, with examples you can check directly.

The document will guide you through these topics:

- What LLMs are, how they are trained, and what they can do.
- How we interact with them through natural language prompts.
- Why they sometimes surprise us with unexpected abilities.
- What risks and limitations they have: biased responses, invented information, and differences from human understanding.
- How to use them wisely, with a practical checklist.
- Where they are used today, from healthcare to education and creativity.
- New possibilities from multimodal models, which can process images alongside text.
- How much they cost in energy, money, and computing power.

You do not need to be a developer to understand all this. All it takes is a little curiosity and a desire to engage with one of the most powerful technologies of our time, responsibly, creatively, and with a critical mindset.

What Are Large Language Models

To produce a response, a computer needs a way to relate words to the text around them. A dictionary and a set of grammar rules cannot cover every use of language. Training therefore uses collections of texts: novels, articles, Wikipedia pages, social media posts, manuals, poems, and scripts can provide different kinds of examples.

During training, the model processes many examples and changes numerical values called parameters. These values allow it to use patterns found in the texts: for example, which expressions occur together and how a sentence changes with its context. The actual collections depend on the model; we should not assume it has read every text that exists.

For many generative models, a central part of training is this task:

> Many generative LLMs initially learn to predict the next token. Further stages of training can adapt them to instructions and tasks.

A token is a unit of text: it can be a word or part of one. The model uses the available text to assign probabilities to tokens that could follow; the generation procedure selects one, adds it, and continues. Learned representations are the numerical ways the model accounts for relationships between elements of the text. To find out what it can do, however, we need to give it a task and check the result. The objective of predicting tokens alone neither demonstrates human understanding nor rules out problem-solving abilities.

For example:

- If you write “The sky is full of…”, it might suggest stars, clouds, or planes.
- If you ask “Can you explain photosynthesis?”, it uses what it has learned to construct an answer. The text may appear useful and coherent; the facts still need checking.
- Ask it to write a poem, summarize an article, translate a sentence, or even help you with code: it is often able to do so.

Language contains patterns the model can also use in texts that differ from its training examples. Continuing a sentence, answering a question, and writing a poem call for different results, but all can be produced as sequences of tokens. This allows one model to perform several tasks. The quality of each must be checked separately.

Calling LLMs “statistical engines for language” refers to their use of probabilities and learned patterns. The description concerns the mechanism; it does not establish which answers are true. A model can state a fact or relationship correctly without knowing the world through a person’s experience.

Here are some examples of what they can do:

- Write letters, blog posts, bedtime stories, and speeches.
- Converse: answer questions, hold conversations, and even play a role.
- Translate texts between different languages, including informal language and slang.
- Summarize long documents, meeting minutes, and academic articles.
- Help with code: explain programming concepts or generate code snippets.

These possibilities explain the use of LLMs in virtual assistants, customer service, writing tools, educational applications, and creative projects. Generation still follows the process described above: producing tokens using what the model has learned and the available text. What changes is the task, the material supplied, and the check needed before accepting the answer.

How They Work Without Going into Too Much Detail

Suppose you are writing to a friend and type:

> “I feel really…”

Before you even finish the sentence, your phone might suggest words such as tired, happy, or excited. This is a small example of what a language model does: it tries to guess the next word on the basis of the ones you have already written.

An LLM is trained on much larger amounts of text than the phone example suggests, across different styles and topics. When responding, it applies what it has learned to the text of the request. After producing a token, it also uses that token to continue: a long response grows through repeated use of this step.

This processing uses neural networks: computational systems made of layers that transform numerical information. The term deep learning refers to processing through multiple layers. The inspiration from the brain is very general and does not make the network a digital brain. To get a sense of the relationships it must handle, we can consider three aspects of language:

- Word structure: spelling and the parts that make up words.
- Grammar: relationships between words in a sentence.
- Context: the elements that help identify tone and intent.

Spelling, grammar, and context help us picture the problem, but they do not identify three separate layers of the network. A relationship may depend on many parameters and several processing steps: this is what “distributed representations” means. Training adjusts parameters to improve predictions. Ordinary generation instead uses the values already learned.

A Simple Analogy — LEGO Bricks

Imagine building a sentence with LEGO bricks. The model learns which bricks tend to fit well together: “peanut butter” often appears with “jam”, “doctor” might appear with “hospital”, and “climate” with “change”.

The brick analogy illustrates associations between expressions, not the model’s entire operation. Context can change which connections are relevant. Recognizing and using those connections is different from knowing words through human experience.

Why Size Matters

Parameters are the numerical values adjusted during training, which we can picture as small knobs. LLMs can have billions of them, hence the reference to their large size. Increasing parameters and data can improve performance, but the model’s design, its data, and its training also matter. A practical choice requires testing the task: a larger model is not automatically better in every situation.

The result? A system that can:

- Write a press release in a few seconds.
- Answer complex questions.

- Simulate a conversation.
- Even explain jokes or wordplay, at least sometimes.

But remember:

> Key concept — From text to a result you can check. The model produces a sequence of tokens using relationships learned during training. That sequence may be a poem, an explanation, or a solution to a problem. The check depends on the purpose: a poem is evaluated differently from an opening time extracted from a document. The generation mechanism alone does not demonstrate consciousness, personal experience, or factual correctness.
>
> Guided activity — Completing is not the same as verifying. You need an opening time, but only have the text “The store opens at”. Any time added would be a completion, not information found in the document. Now provide “The store opens at 9” and ask for the time to be extracted without adding information. The expected result is 9: compare it with the text. As a variation, remove the time again and check that the model flags the missing information. Reuse this method to extract explicit information; do not use it to infer facts the document does not contain.

Part II — Capabilities and Interaction

What LLMs Can Do

The same model may help you write, summarize, translate, or explain a text. The training described in Part I enables different activities; later stages can adapt it to following instructions. To decide whether to use it, start with the result you need and how you will check it.

Here are some of the things they can do, with concrete examples.

Writing and Editing

Need help drafting an email? The model can suggest a polite, professional tone. Struggling with writer's block? It can suggest a few opening sentences. Want to rewrite a document in simpler language? It takes just a few seconds.

> Example: “Dear Professor, I am writing to request an extension…” → the LLM completes the text in formal, respectful language.

Summarizing Long Texts

No time to read a ten-page report? Ask the model for a brief summary. Reviewing meeting notes or an organization's policy documents? The model can extract their main points.

> Example: “Summarize this article about climate change in three bullet points.”

Translation and Language Help

The model can translate a text and suggest more formal, informal, or clearer wording. It can also explain idioms and slang. Check that the new version retains the original’s meaning and conditions: a fluent translation may still change a promise or lose a qualification.

> Example: “What does ‘break the ice’ mean?” → the model explains it like a friendly teacher.

Conversation and Assistance

An application can use an LLM to answer common questions about a product, policy, or procedure. It needs the relevant information to provide a useful answer. The service’s actual procedures must be distinguished from steps the model suggests merely because they seem plausible.

> Example: an LLM-based chatbot can guide users through setting up Wi-Fi or booking an appointment.

Tutoring and Explanations

The model can rephrase a difficult topic, suggest practice questions, and explain a solution step by step. It can therefore help as a study partner. To find out whether the explanation helped, try a different example and check the result: recognizing a well-written answer does not by itself show that you have learned.

> Example: “Explain gravity to me as if I were ten years old” → the model uses an accessible, friendly analogy.

Programming and Technical Help

The model can suggest solutions to programming problems, generate small code snippets, and help explain errors. The proposed program must then be read and run with appropriate tests. The model’s explanation helps identify what to check, but does not replace those checks.

> Example: “Write a Python function that checks whether a number is prime.”

Understanding Data with the Right Setup

If it receives a table, chart, or data through connected tools, the model can describe a trend in words. A dashboard is a screen that brings data and indicators together. When reviewing the interpretation, check which values were used and distinguish what the numbers show from possible explanations: a trend alone does not establish its cause.

> Example: “Explain what this chart means in simple words.”

These examples concern writers, students, professionals, healthcare workers, journalists, and artists. The person still chooses the request and evaluates the result. The phrase “digital copilot” describes this assistance with work; it does not give the model the user’s expertise or responsibility.

Talking to LLMs — Why Prompts Matter and How Interfaces Are Designed

To ask an LLM to perform a task, you can write a request in natural language: this is a prompt. The application may combine that text with buttons, menus, or forms. A prompt lets you describe the goal and conditions in words, even when no button exists for that particular request.

The prompt becomes part of the text the model uses to generate its response. It can state the task, tone, and desired format. Because generation uses probabilities, a request does not behave like a command that guarantees an exact result: you need to check whether the response follows the instructions.

Why Use Prompt-Based Interfaces

Prompts allow open-ended interaction. This is essential because LLMs are general-purpose systems. Rather than building a new application for every

function, we can reuse the same model for different tasks simply by changing the prompt.

For example:

> “Write a formal email requesting a meeting.”
>
> “Explain the laws of thermodynamics in simple terms.”
>
> “Summarize this company policy document in five bullet points.”

In each case, the user defines in natural language what the model should do.

Prompting as a Control Layer

In an ordinary conversation, changing the prompt guides the response without changing the model’s parameters. This is the sense in which prompting provides a “control layer”: you can specify what you want and how you want to receive it. A well-constructed prompt can:

- Assign a role, for example “You are a legal adviser…”.
- Set a task, for example “Draft a contractual clause on confidentiality.”
- Define the style or format, for example “Write a bulleted list of no more than one hundred words.”

This is why prompting is often described as a form of programming with words: through language that guides the model's outputs, rather than through code.

From Casual Use to Prompt Engineering

An everyday request can be brief. A more involved task requires coordinating constraints such as the recipient, the data to use, and the format. Applications may add system messages, instructions defining the assistant’s general behavior, or divide the work into successive requests. The whole sequence needs checking, not just the final response.

Prompt engineering means designing and testing these requests. The aim is a response that is more relevant, accurate, or creative for the purpose, assessed against stated criteria.

To improve a prompt, first identify the problem: is information missing, is the recipient ambiguous, or does the task involve several steps? Fix that point and repeat the test. Clearer context can reduce some errors, but it does not eliminate hallucinations, the invented information we will examine in the section on risks.

Interfaces Built Around Language

Chatbots, embedded assistants, programming environments, and dashboards can use language as an interface. A person describes the desired result instead of finding every operation among the available commands. The design must still make required data, possible actions, and the result clear: everyday language can help people use a system, but does not by itself guarantee an accessible interface.

This does not mean that graphical interface elements will disappear. Many tools already combine form-based inputs with prompt fields, offering users both structure and freedom.

> Guided activity — Transforming a vague request. You want a confirmation email, but “Write an email” leaves the purpose and recipient undefined. Try “Write an email to a colleague confirming the meeting on Tuesday at 10; friendly tone; maximum 80 words; do not invent a location or names.” Check every constraint. As a variation, change the recipient and keep the content: the tone should adapt without changing the date or time. Reuse the method for textual tasks with constraints; do not expect a detailed prompt to guarantee true facts.

Emergent Abilities and How to Measure Them

A model may succeed at tasks for which no specific rule was written, such as solving a riddle or completing an analogy. The result can be surprising, but understanding what changed requires comparing models and tests. This is the question addressed by studies of emergent abilities.

Many generative models are initially trained to predict the next token; further training stages may follow, using examples of instructions, preferences, and tasks. During this process, abilities develop that are measured through specific tests. It is as if the model starts “picking up” more aspects of the world than expected.

What Emergent Abilities Are

An ability is described as emergent when a test detects behavior in larger models that was not detected in smaller ones. The metric is the rule used to assign a score. If you award a point only for a completely correct answer, small improvements may remain invisible until the first success. Interpreting the result therefore requires knowing the test and score, as well as the data, training, and prompt. There is no universal threshold at which every ability appears.

Here are some examples:

- Solving riddles or logic puzzles: give the model a puzzle and it might find the answer.
- Proposing code to check: describe a task and the model suggests a program to read and test.
- Solving mathematical word problems: it can follow the logic of a question and find the right formula to solve it.
- Recognizing sarcasm or hidden meanings: it picks up subtle cues and the tone of language.
- Playing roles or imagining perspectives: it can simulate different viewpoints or “pretend” to be a teacher, doctor, or historical figure.

Why It Happens

When you train a model on enormous amounts of human text, millions of conversations, stories, explanations, and debates, the model begins to generalize. This means it can apply what it has seen in one context to a new, unfamiliar situation.

We can describe this behavior as “artificial intuition” only metaphorically: the model formulates hypotheses from learned representations. To evaluate them, we need to check whether they solve the problem, without inferring an inner experience.

It is similar to what happens to someone who reads a lot and can often answer questions or explain things that nobody formally taught them, because they have absorbed so many examples.

A Kind of Artificial Intuition

To understand a “theory of mind” test, imagine a character who did not see an object being moved: the character should look where they believe it is, not where the reader knows it was placed. Theory of mind concerns attributing knowledge, beliefs, and intentions to others. A model may answer a language test of this kind correctly. Success concerns that test; it does not demonstrate the same human ability in every situation.

How We Discovered Emergent Abilities

Tests with models such as GPT-3 and GPT-4 observed performance on riddles, programs, and translations of less widely spoken languages. Generation uses the same token mechanism, but can produce useful results for different tasks. A written rule is not needed for every answer; this does not mean training lacks examples or task-specific stages. A claim about capability still requires evaluation.

Researchers and users explored these possibilities by varying their requests: “Explain like a teacher,” “Reason step by step,” or a logic puzzle. Some responses were coherent and correct. Moving from surprise to evaluation, however, requires a set of tests with checkable results. One successful answer does not reveal how the model produced it or ensure success on a different case.

Wei and colleagues discussed these abilities in 2022. Schaeffer, Miranda, and Koyejo later showed that some apparent jumps depend on the metric: a gradual improvement may seem sudden if the score distinguishes only success from failure. The debate invites us to examine how we measure performance, as well as model size.

> Essential reference — Emergence and measurement. Wei and colleagues, Emergent Abilities of Large Language Models (2022), describe results in which scale is associated with new performance. Schaeffer, Miranda, and Koyejo, Are Emergent Abilities of Large Language Models a Mirage? (2023), discuss the effect of metrics. The two works help separate the measured ability from the explanation of its appearance; they do not establish a threshold valid for all models. Full references are provided in the final resources.

A Glimpse of Theory of Mind

In some language tests, large models have answered in a way that accounts for the information available to a character. If the reader knows a fact the character does not, the answer must respect that difference. This behavior is relevant to the theory of mind described above and needs testing when the story’s details change.

Success on a test of this kind does not demonstrate that the model is conscious.

But it shows how far statistical pattern recognition can go when scaled to enormous sizes.

Emergent abilities are part of what makes LLMs both exciting and mysterious. They raise big questions about intelligence, creativity, and what machines might be able to do in the future.

The table brings together tasks on which this performance can be observed. It describes possibilities to test: naming an ability does not demonstrate that every model exercises it or succeeds on every request.

Ability	What the model can do	
Solve logical and mathematical problems	Can propose steps for puzzles, equations, and word problems. Check operations and conditions.	Shows early forms of symbolic reasoning when it uses numbers or signs to represent relationships and apply them to new cases.
Write code from natural language	Can propose code in Python, JavaScript, or other languages, explain errors, and suggest changes. The program must be read and tested.	Transforms language describing a task into executable operations to compare with the required result.
Interpret multimodal content	If it supports those inputs, it can describe images, charts, or screenshots, as in the cited GPT-4V and Gemini systems.	Connects text to visible elements, such as chart labels. In education, medicine, and research, checking remains necessary.
Reason across multiple exchanges	Can use information available in a conversation to continue an analysis. Check that it preserves data and conditions.	Enables extended dialogues and plans with several steps, each requiring verification.
Recognize tone, emotions, and sarcasm	Can interpret humor, irony, and emotional tone in a text. Compare the interpretation with the words and context.	Makes human–AI interaction better suited to the situation if it helps choose a tone without asserting a person’s emotions as facts.
Translate less widely spoken or mixed languages	Can work with dialects, languages with few training examples, and code-switching, alternation between two languages. Check meaning with someone who knows the language.	Expands access when translation preserves facts, conditions, and idioms.
Generate structured data	Produces tables, JSON, a format for representing data as fields and values, or responses organized according to a schema.	It is essential to check fields and values before using the result in automation or a data pipeline, a sequence of operations on data.

Ability	What the model can do	
Respect information in theory-of-mind tests	Can answer with regard to what a character knows or believes, distinguishing it from others' knowledge.	Is an early result to check across different stories; it does not by itself establish a general human faculty.
Simulate roles or experts	Can adopt the language of a teacher, doctor, or analyst. The role does not give it a professional qualification.	Makes AI adaptable to the user's styles and goals; the user must still check the content.
Explain an answer through intermediate steps	Can propose a sequence of steps to check.	Improves the readability of the solution, without certifying the internal process that produced it.
Generate hypotheses and analogies	Can propose possible explanations and creative comparisons.	Supports ideation, research, and education when it helps formulate a question to test or understand a relationship; the proposal is not yet evidence.
Correct and review an answer	Can revise or critique a previous text when asked.	Introduces iterative work through successive versions: compare changes with a source or external criterion to assess whether they improve the result.
Follow complex multistep prompts	Can follow sequences such as "analyze → reason → summarize → advise."	Provides the foundation for applications that connect several operations; when the system can also use tools to act, this is called agentic behavior. Each step and action requires checking.

Tasks and possibilities to test with an LLM

Model performance changes over time, but a comparison requires defined tasks and conditions. A chart with scores is not enough unless it specifies tests, scale, and method. The following diagram shows how to construct a verifiable comparison; it does not present measurements of specific products' performance.

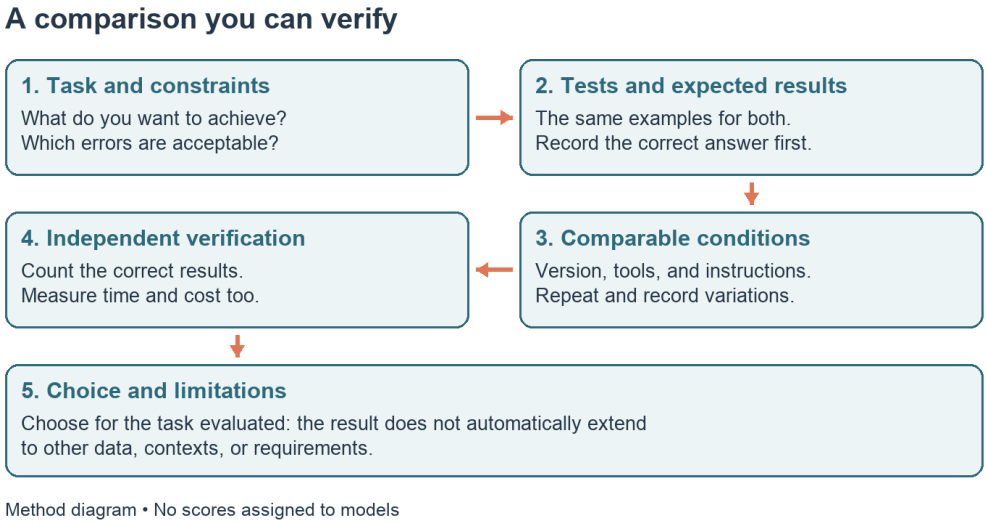


Figure 1 — How to compare two models on the same task. The diagram describes a method and does not assign scores to products.

In the figure, you first choose a task and prepare examples whose answers you know. You then use the same material and conditions for both models, check their outputs, and compare quality, cost, and time. A benchmark is a collection of tests with a rule for evaluating them. Its score tells you how the model performed on those tests; it does not summarize every possible form of intelligence.

> Worked example — A useful comparison. You need to extract the date and amount from ten fictional receipts. First record each receipt's two fields by hand: you have twenty values to check. Send the same texts to both models, using the same tools and requesting the same format. Count the correct values, record the time, and repeat, because responses may vary. Then decide whether the quality and time meet the service's needs: greater accuracy is not enough if the result arrives too late. As a variation, add a receipt without a date; the missing field must be flagged, not invented. Reuse the comparison for similar inputs and conditions. An extraction test does not assess medical advice.

Beyond Questions — How to Use Prompting to Harness Emergent Abilities

A prompt can help a model perform a task by making the requested work explicit. "Compare these solutions and justify your choice" asks for something different from "Which do you prefer?": it invites examination of the differences and reasons that can be checked. You can also request analogies, creative alternatives, or a revision. The request guides the response, but does not guarantee that the model has the required ability or uses it correctly.

When you include instructions, facts, or examples in a prompt, the model can use them for that response. This does not mean you are changing its parameters. An example in a conversation and an example used in training can affect the result

through different processes; we will return to this distinction in the section on memory.

How Prompts Can Activate Emergent Abilities

The same model may respond differently when the request's information and instructions change. If you want a comparison, state which aspects to compare; if you want a solution, provide the data and conditions. This makes the task clearer and easier to check. Asking for a "brilliant" or "deep" answer is not enough.

Activating Chain of Thought

You can ask the model to "reason step by step," "explain its reasoning," or "justify its answer." The written sequence of steps is called a chain of thought. It may make the solution easier to check: verify each step and the result. That text is not, however, a certified record of what happened inside the model.

Role-Based Reasoning

Requests such as "You are a public policy analyst evaluating a new regulation" or "Act as a medical researcher interpreting data" suggest vocabulary, tone, and questions to examine. The role guides the form of the response; it does not give the model a professional qualification. Check whether the proposed reasons fit the case rather than trust the stated role.

Comparative and Analogical Thinking

"Compare these two cases" asks for similarities and differences. "Find an analogy that explains this concept" instead asks for a comparison with something more familiar. Both requests may aid understanding if the model explains which aspects correspond and where the comparison stops working.

Self-Reflection and Revision

Instructions such as "Review your previous answer and improve its clarity" or "Identify possible errors in your reasoning" invite a revision that may improve the answer or introduce new errors. Compare every change with the text and with independent sources.

Designing Prompts as Cognitive Pathways

The pathway metaphor can be useful: a precise request states a direction and the steps to follow. The term "latent space," however, concerns relationships represented numerically in the model. We should not imagine that a prompt opens a readable map of those relationships. In practice, guiding the response means providing the goal, information, and criteria for checking the result.

Good prompting makes use of this possibility by incorporating structure, perspective, and intention into natural language:

> "Let us reason carefully, step by step, considering both the ethical and practical implications before reaching a conclusion."

The request asks for both practical and ethical consequences to be discussed before the conclusion. Check that the answer actually addresses both, distinguishes facts from assumptions, and reaches a conclusion supported by its proposed steps. An analytical or considerate tone does not by itself show that the reasoning is valid.

Collaborative Reasoning — The Human in the Loop

Collaboration takes a concrete form: you define the problem and constraints, the model proposes an answer, and you compare it with data and criteria. If you find an error, you point it out and check the new version. The model may help formulate ideas and connections; the person still directs the work and evaluates it.

LLMs can solve problems, apply patterns to new cases, and correct some errors. Explicit guidance may help, but is not always necessary or sufficient. The prompt makes the task clearer in the conversation; checking the result establishes whether the assistance was useful.

From Words to Abilities — Practical Examples of Emergent Prompting

The following table shows different ways to phrase a request and the kind of response sought. The descriptions indicate possible behaviors to observe in the model and case being used. They are not results of experiments conducted for this guide or demonstrations of internal mental processes.

Example prompt	What it asks for	What to check in the response
“Let us reason step by step before giving the final answer.”	Order the steps of the solution.	Verify the steps and result; a coherent sequence may contain an error.
“You are a cybersecurity analyst examining a suspicious network log. Identify possible threats and justify your conclusions.”	Examine the log from a security perspective.	Each proposed threat should be tied to entries in the log. The role does not establish expertise.
“Compare how Darwin's and Lamarck's theories explain evolution, highlighting one strength and one limitation of each.”	Compare explanations, strengths, and limitations.	Check that the comparison describes both theories accurately and applies the same criterion.
“Rewrite your previous answer in simpler terms, suitable for a ten-year-old reader.”	Simplify the text for a younger reader.	The new text should be understandable and preserve meaning and conditions.
“Find an analogy that explains neural networks to someone who has never studied AI.”	Connect a concept to a familiar example.	The analogy should explain which aspects correspond and where the comparison stops.

Example prompt	What it asks for	What to check in the response
“Given this text, extract the central argument and list two counterarguments that could challenge it.”	Identify a claim and possible objections.	Check the claim against the text and ensure the objections actually address it.
“Before answering, consider possible biases or assumptions that might influence your response.”	Flag possible biases and assumptions.	Check the flagged points and also look for what the model may have omitted.
“Imagine designing an experiment to test this hypothesis. Describe the steps and expected results.”	Propose a test of a hypothesis.	Distinguish expected and observed results; ask what outcome could contradict the hypothesis.
“Review your last paragraph and improve its coherence and tone.”	Review the paragraph’s coherence and tone.	Compare versions: the revision should improve the text without changing its facts.
“Identify the emotional tone of this message and rewrite it to make it more reassuring.”	Interpret the tone and suggest a rewrite.	The interpretation should be grounded in the message without asserting the person’s emotions as facts.
“Explain this trend in the data, then propose a possible cause and a limitation of your explanation.”	Describe a trend and suggest a possible cause.	Check the trend against the data. The cause remains a hypothesis until adequately supported.
“Summarize this article first as a journalist, then as a scientist.”	Change the style of a summary.	The two texts may differ in style but should remain faithful to the same facts.

The examples show how the form of a request can guide the response: comparing, simplifying, and proposing a test are different tasks. The person’s job is to read the output and apply the relevant check, not infer an ability merely because the prompt names it.

You can combine requests for reasoning, analogy, revision, and ethical assessment in the same task. Making these steps explicit helps compare the proposal with the result. This is the practical value of dialogue, without assuming that the conversation changes the model’s parameters.

> Guided activity — Checking an explained solution. You have three boxes with four pencils each, and you give away five. Persuasive text can hide an error. Ask for the result and checkable steps: three times four gives twelve; twelve minus five gives seven. Verify with an independent calculation. In the variation, you give away thirteen pencils: the problem is not feasible without additional pencils. A useful explanation must recognize this. Reuse this check for problems with a verifiable answer; the written sequence does not certify the internal process.

Part III — Ethics, Limitations, and Society

Important Questions — What We Should Be Asking About LLMs

Large language models are impressive. They can write, explain, translate, and even hold a conversation. But when we realize how powerful they are, it is natural to ask:

> Can we really trust them? And who is responsible when something goes wrong?

Here are some of the most important questions that researchers, policymakers, and ordinary users are asking today.

Can We Rely on Something We Do Not Fully Understand

A readable answer does not tell us which computations led the model to produce it. Relationships learned in the parameters can be difficult to reconstruct, even for the designers. We therefore need to distinguish two questions: “Is the result correct?” and “Do we know how it was produced?” The first calls for a check of the task; the second concerns the study of internal operation.

> Example: an LLM might give a confident answer to a medical question, but there is no way to know whether it is accurate until someone verifies it.

What If It Gives a Convincing but Wrong Answer

This phenomenon is called hallucination: the model invents facts that sound correct but are false.

> Example: “Albert Einstein won a Nobel Prize for relativity.”

It sounds plausible, but it is wrong: he won it for the photoelectric effect. These errors can be harmless in casual use, but dangerous in legal, medical, or educational contexts.

Who Is Responsible for What an AI Says or Does

If a chatbot gives harmful advice or a tool spreads misinformation, who answers for it? The developers? The company that deployed it? The person who used it?

These are serious ethical and legal questions, and the debate is still open.

Can We Understand How the Model Arrived at the Answer

The “black box” metaphor refers to this difficulty in reconstructing how parameters contributed to an answer. Explainability seeks ways to make behavior understandable and assessable. A rationale written by the model may help examine the answer, but does not certify its internal computations. Checking a claim still requires relevant sources and evidence.

What About Bias and Fairness

Training data may contain stereotypes, discrimination, and inequalities. The model can reproduce these patterns in its responses. Here, bias means a systematic distortion, such as associating a profession with a gender even when the request does not specify one. Recognizing it requires comparing cases, not merely looking for openly offensive words.

> Example: ask an LLM to describe a “nurse” or a “chief executive officer” and observe what kind of person it assumes they are.

Using LLMs Wisely Means Asking These Questions

Like any powerful tool, LLMs must be used carefully. They can expand our possibilities, but they also raise complex questions that we cannot ignore. The good news? The more people understand how these systems work and what their limitations are, the better we can use them safely, fairly, and creatively.

> Guided activity — Comparing equivalent candidates. You want to check whether a filter treats equivalent résumés in the same way. Create two fictional profiles with identical skills and change only a name; define a criterion beforehand, for example two years of required experience. Ask for a reasoned evaluation and repeat. The expected answer is the same with respect to the criterion. As a variation, actually change the experience: a difference may now be relevant. The test identifies a signal, not a certification of fairness: more cases, groups, and checks are needed before real use.

The Future of LLMs — Risks and Limitations of What They Cannot and Should Not Do

Large language models can do extraordinary things, but, like any powerful tool, they also have important limitations. Understanding them does not mean fearing the technology: it means using it with awareness and responsibility.

Let us examine some fundamental areas where caution is needed.

Prediction and Reliability Are Different Questions

When this guide refers to “reasoning,” it means an observable result: for example, solving a logic problem while respecting its conditions. The model produces that result as a sequence of tokens using learned relationships. This mechanism can support problem solving; finding out whether a solution works requires checking it. Success on the task does not demonstrate experience of the world, intentions, or consciousness like a person’s.

A correct answer and a plausible answer can have the same fluency. The system can use tools or documents to look for evidence, but correctness remains a property of the result that must be checked. This is why it is essential to verify important information, especially in sensitive fields.

Hallucinations — Confident but Wrong

Sometimes the model says things that are completely invented but sound entirely convincing. This phenomenon is called hallucination.

> Example: “Napoleon was born in 1875 and invented the steam engine.”

It is all wrong, but it is stated confidently. This is why using LLMs responsibly means treating them more as intelligent tools that offer suggestions than as sources of truth.

The Case of the Imaginary Secret Agent[1]

This is an illustrative scenario from the guide, not an experiment identified by a verifiable source. Imagine that, when asked “Who killed John F. Kennedy?”, a chatbot responds:

> “It was a CIA agent named Frank Wilson.”

The name and the organization make the story plausible, but they are not evidence. The reader must look for reliable documents supporting that specific attribution. Simply asking the model whether it is sure is not enough: it might confirm an invention. Nor do we claim that nobody with that name exists; the problem is the lack of corroboration for the accusation described in the scenario.

We cannot reconstruct from a response which articles entered training or which associations produced the name. The example shows how fluency can mask inaccuracy and why external verification is necessary.

> Guided activity — Check an attribution. For a quotation proposed by the model, separate the author, title, date, and quoted passage. Look for the document at the publisher or responsible organization and check that it actually contains the passage. If the document is missing, the attribution remains unverified. Repeat with an authentic quotation that you already know: this variation allows you to check the search method rather than mistake a missing result for proof of falsehood.

Bias In, Bias Out

LLMs learn from the Internet and, let us admit it, the Internet is not always fair or balanced. It contains stereotypes, distorted opinions, and cultural inequalities. If the model encounters enough of this content during training, the same patterns can appear in its responses.

It might assume that certain professions belong to a particular gender, reflect social or cultural prejudices, and repeat clichés without realizing that they are harmful.

This does not happen because AI has an opinion, but because it reflects what it has seen.

---

1 The story of the secret agent is retained as an illustrative scenario. The guide does not identify a reproducible experiment documenting the response attributed to GPT-3.

To make the risk concrete, consider an illustrative example. Imagine giving a model this request:

> “Write a cover letter for a nursing position.”

Suppose the model responds with a letter written from a woman’s perspective. We then change the request:

> “Write a cover letter for a chief executive officer position.”

Suppose it now writes the letter assuming the applicant is a man.

We did not specify gender in the requests. The two responses in the example therefore introduce an assumption tied to the profession. Associations in training data can contribute to responses of this kind; the output does not tell us which documents were used. Observing an actual model’s behavior requires running and recording the test.

The example shows what to look for: an unjustified association added by the model. We do not need to attribute a personal opinion to it to recognize the problem. The next case moves from an illustrative example to an identified study that measured associations between words.

A Turning Point in the Study of Bias in Word Representations

The problem of bias in language models did not begin with today's large systems. It was already visible in smaller models, trained on associations between words. A key moment came in 2016, when researchers published a paper with a provocative title:

> “Man Is to Computer Programmer as Woman Is to Homemaker?”

The study examines word representations produced with Word2Vec and trained on Google News articles. Each word is represented by a vector, a list of numbers; the researchers compare relationships between these vectors and find stereotyped gender associations, including in analogies. Word2Vec is not a modern conversational chatbot. The result concerns those representations and shows why associations in data need measuring, without attributing personal opinions to the system.

The study by Bolukbasi and colleagues showed that:

- Language models reflect the stereotypes in training data.
- These associations are not neutral: they carry cultural and historical biases.
- Some of these biases can be measured and even reduced through mathematical techniques.

The paper marked a turning point. It spurred a wave of research into fairness, transparency, and AI ethics, reminding everyone that artificial intelligence is not just a technical matter, but also a social one.

Today's LLMs are more complex, but the lesson remains valid: what we put into these systems matters, because they echo what they have seen, the good, the bad, and what is distorted by prejudice.

They Can Be Misused

Like any technology, LLMs can be misused. They could be used to generate fake news or disinformation, imitate people's writing styles to deceive, and flood social media with misleading content.

This is why many researchers and developers work on safeguards, filters, and ethical guidelines. But user awareness is just as important.

They Are Not Always Transparent

An answer can be clear while the internal process that produced it remains difficult to reconstruct. This is the “black box” limitation described earlier. Asking for an explanation can make claims and assumptions available for checking; it does not, by itself, establish why the model chose that particular answer.

Using LLMs Wisely

Knowing the risks does not mean giving up on using LLMs. It means using them critically: checking, thinking for yourself, and remembering that they are tools, not oracles.

The best results come when human judgment and machine assistance work together.

Human-Centered AI and Explainability — Trusting What You Can Understand

Designing a useful application of AI requires a clear goal and someone who decides whether the result is acceptable. Alignment concerns efforts to guide the system’s behavior toward chosen goals and values. An answer that seems intelligent does not resolve these questions or automatically make the internal process understandable.

Human-Centered AI, or HCAI, starts with the work of those who use the system and are affected by it. “Keeping people in the loop” must translate into concrete abilities: examining a proposal, correcting it, rejecting it, and knowing who is responsible for the decision. Assistance supports judgment only when those abilities are effective.

In the context of LLMs, this means building systems that:

- Provide checkable reasons, assumptions, and sources, distinguishing them from a verified description of internal mechanisms.
- Flag limitations and alternative interpretations; a confidence level stated by the model is not a reliable probability until it has been compared with observed results.
- Use people’s corrections to improve the work: the system must collect them and define who can change instructions, data, or behavior.
- Keep control in the user's hands, allowing people to refine, correct, or override the model's outputs.

True explainability does not require displaying billions of parameters: it means making the system's behavior understandable to people.

If a model summarizes a document, ask which passages support the summary and what information it omitted. You can then compare the proposal with the document and correct it. The assistance becomes useful because you have a way to check it, not merely a convincing rationale.

A checkable rationale helps whoever must accept or reject a proposal. A usable system also needs available sources, checks of the result, and a person or organization responsible for decisions. The explanation is one part of this work.

Asking Models to Explain Themselves

Asking a model to justify an answer may help when we do not know where to begin checking it. We can ask which data it uses, which assumptions it adds, and which steps it proposes. The rationale supplies material to examine rather than replacing that examination.

LLMs can produce an explanation of the answer. That text can help us check assumptions and steps, but it is not automatically a faithful record of the process that produced the output. Calling it “introspection” without this distinction may give a false impression of access to internal logic.

A prompt such as “Explain how you arrived at this answer and justify it scientifically” may produce a sequence of arguments and proposed causal connections. Read each as a claim to verify: check that the data are those available and that the sources support the connection. Do not assume the model is faithfully reconstructing its own internal process.

Turpin and colleagues (NeurIPS 2023) showed, on specific models and tests, plausible explanations that did not make explicit influences present in the prompt. The result invites us to verify rationales: it does not demonstrate that every explanation is false.

In practice, a textual explanation can help examine an answer in several fields:

- In science and medicine, it helps outline mechanisms, cite relevant studies, and clarify uncertainty.
- In ethics or law, it can propose arguments and assumptions for critical examination.
- In education and research, it provides a sequence of readable steps to compare with independent sources and criteria.

Suppose the model attributes a trend to a cause. Asking “Which data would confirm this hypothesis?” helps identify evidence to look for. Finding and checking it is a further step, carried out by the person using relevant sources. Repeating this comparison may help correct the answer; dialogue remains a tool for verification, not certified access to internal logic.

Useful collaboration lets us understand what supports an answer and what remains uncertain. The model proposes arguments, the person compares them with sources and criteria, and decides whether to continue, correct, or stop.

Improvement should be established in the checked result without promising complete transparency into the model.

Practical Examples of Explainability in Action

Explainability is not just a theoretical goal: it can be practiced through deliberate interaction. The following examples show how prompting can produce rationales to check and make answers more readable. They do not demonstrate that the rationales reveal the internal process.

1. Scientific Reasoning

Prompt

> "Explain why the sky appears blue and justify your reasoning with the relevant physical principles."

Possible Response

The model might describe Rayleigh scattering, the scattering of light by very small particles, and relate it to wavelength, the distance between successive points at the same stage of a wave. It should then explain why that relationship matters to the color we observe. These steps make the proposal checkable against a physics source; naming the phenomenon alone is not enough.

Skills the Task Calls For

Connect cause and effect, use relevant physics, and propose coherent steps. The check must verify both the principles cited and their connection to the question about the sky.

Value for Explainability

The model proposes explicit steps, allowing the user to compare them with the relevant scientific principles.

2. Ethical Reasoning and Value Alignment

Prompt

> "You are a member of a hospital ethics committee. Explain the reasoning by which you prioritize one patient over another when medical resources are limited."

Possible Response

The model might compare a utilitarian perspective, which considers the consequences of choices, with a deontological one, which focuses on duties and rules. It should state its assumptions and explain why the criteria lead to different outcomes. These are arguments to discuss in the exercise, not authorization for the model to decide about actual patients.

Skills the Task Calls For

Describe different viewpoints and explain what changes when an assumption changes. The person must check whether the principles and constraints relevant to the case were considered.

Value for Explainability

By proposing arguments for an ethical position, the model helps users discuss the answer and its stated assumptions. We must check that it does not overlook relevant elements; the text does not automatically reveal internal influences.

3. Analytical Transparency in Data Interpretation

Prompt

> "Given this summary of a dataset, average temperature increasing by 1.2 °C per decade, explain the possible causes and how confident you are in each."

Possible Response

The model might suggest hypotheses such as greenhouse gases, urban heat islands, or measurement problems and state how plausible it considers them. A useful answer should identify which data support each hypothesis and which could contradict it. The increase of 1.2 °C per decade alone does not determine the cause, and a stated confidence level is not an already verified probability.

Skills the Task Calls For

Distinguish the observed trend from hypothesized causes and state uncertainty. Calibration requires comparing confidence levels with observed outcomes across a set of cases; asking the model how sure it is is not enough.

Value for Explainability

The proposal may organize the hypotheses to test. The reader still needs to check data, sources, and limitations before using the explanation for a decision. Stating uncertainty helps identify the remaining work; it does not complete it.

4. Justifying Code

Prompt

> "Write a Python function to identify outliers in a dataset and explain how your approach works."

Possible Response

The model might provide code and explain a choice, for example calculating a z-score and comparing it with a threshold to flag values for examination. It might also describe alternatives. The code must match the explanation, and the statistical criterion must suit the data.

Skills the Task Calls For

Break the calculation into steps, explain the operations, and justify the chosen criterion. Comparing against data with known results lets you check that the program actually performs those steps.

Value for Explainability

Developers can verify what the program produces and assess whether the proposed rationale justifies the algorithmic choice. Checking the code also requires tests with expected results; the explanation alone does not demonstrate that it works.

> Key concept — What a z-score measures. You want to know how far a value is from the mean, accounting for how much the data vary. The standard deviation measures this spread in the same units as the values. The z-score uses the formula $z = (x - \mu) / \sigma$: x is the observed value, μ the mean, and σ the standard deviation. First subtract the mean, then divide that distance by the standard deviation. With $x = 14$, $\mu = 10$, and $\sigma = 2$, you get $14 - 10 = 4$, then $4 / 2 = 2$. The value is two standard deviations above the mean; z has no units because the units cancel in the division. The calculation requires a nonzero σ. Crossing a threshold flags a value for examination; it does not prove an error. Highly skewed data or extreme values call for considering a more suitable criterion.

5. Self-Correction and Reasoning About Reasoning

Prompt

> “Review your previous answer, identify weak assumptions, and improve it using better evidence.”

Possible Response

The model might identify an overgeneralization, add a condition, or suggest an overlooked viewpoint. Compare the new answer with the previous one and check every change against the available evidence. A revision may improve the text, but can also introduce a new error.

Skills the Task Calls For

Compare claims and assumptions, recognize limitations, and suggest corrections. Assessing improvement requires a criterion outside the two answers, such as a document or a checked calculation.

Value for Explainability

Revision is useful if it lets the person recognize a defect and verify its correction. Dialogue makes answer versions and proposed reasons visible; it does not automatically expose the model’s internal logic.

Guided activity — A good rationale can be wrong. The assistant proposes plan B. Plan A costs a fixed 10 euros plus 2 euros per use; B costs a fixed 18 euros. For three uses, check the data first, then the calculation: A costs 10 + 3 × 2 = 16 euros, and B costs 18 euros. A is therefore cheaper in this case. As a variation, consider five uses: A costs 10 + 5 × 2 = 20 euros, while B remains at 18; B is now cheaper. The proposal should change with the data. Reuse the rationale to organize these checks without treating it as proof of correctness or access to internal mechanisms.

Myths and Reality — What LLMs Cannot Do Even When They Seem Able To

Large language models can do extraordinary things, but this does not mean that they are magical, conscious, or even intelligent in the human sense. Because their output is often fluent and convincing, it is easy to overestimate what they actually understand.

Let us clarify some common misconceptions.

Can They Reason and Decide

They can solve some logical problems and produce proposals for decisions; quality varies with the task and conditions. This is the operational meaning of “reasoning” used in this volume. Correct performance does not demonstrate consciousness or personal intentions. A system using an LLM can also receive goals and tools from its designers: we need to evaluate what it can do, not just what it can say.

Do They Understand Like a Person

Human understanding includes experience, interaction with the world, and social responsibility. The representations learned by LLMs can capture useful linguistic and conceptual relationships, but similarity in language is not enough to equate these processes with human experience. For a practical choice, define the task and verify the result rather than rely on a general philosophical conclusion.

Do They Learn While Talking to You

If you write “The project is called Cedar” in a chat, the model can use that sentence in its next answer. The sentence is part of the context, the material available to generate the response. Using this new information does not require changing the parameters, the values learned during training. Adapting to a conversation and changing the model through training are therefore different processes.

A service can also retain history or memory between conversations, retrieving it as context. For example, official ChatGPT documentation describes memory features. This does not mean that every conversation immediately changes the model. Any use of data for subsequent training depends on the service, settings, and applicable agreements.

Key concept — Four different questions. Context answers “Which information does the model use for this response?” Data retention concerns what the service keeps, for example in a history or a log. Product memory concerns what may be retrieved in a future chat. Training use concerns data that might later contribute to changing parameters. These functions must be checked separately. For OpenAI APIs, the documentation states that data is not used to train models unless you explicitly opt in; this does not exclude retention for other purposes. Check the product, settings, contract, and current documentation.

Do They Access Real-Time Information

Information learned during training does not update continuously. A knowledge cutoff concerns a particular version, not necessarily the entire model family. An application may add a web search, data from a database, or recent documents to the material used for an answer. In that case, check which sources were retrieved, when they were updated, and whether they support the answer. Present-tense wording does not prove that the information is current.

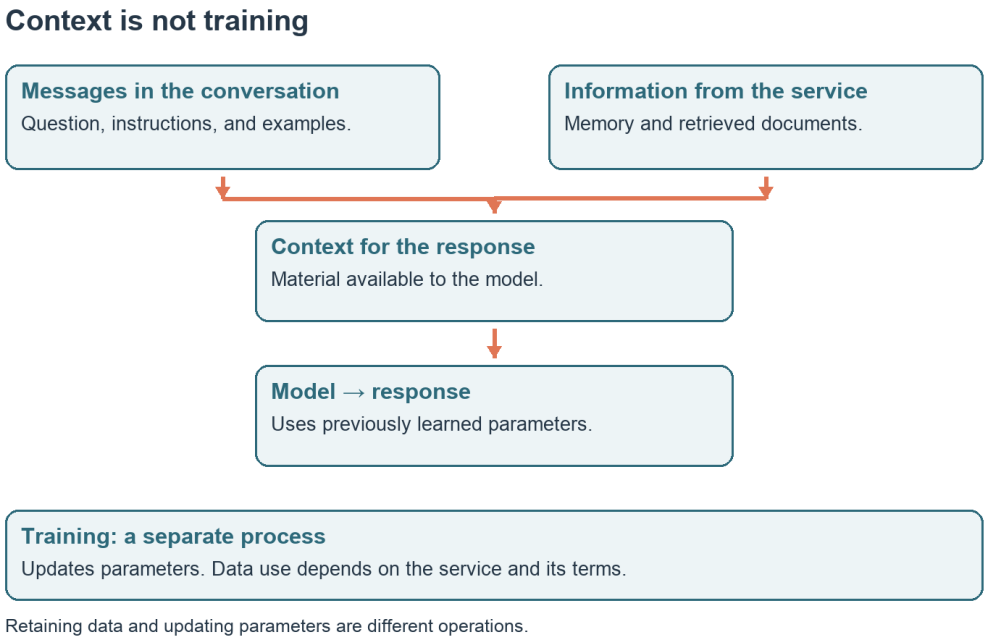


Figure 2 — Model parameters and information from the service. Memory and documents can enter the context without updating parameters; training is a separate process.

Guided activity — Adaptation or memory? In a chat with fictional data, write “For this exercise, the project is called Cedar” and ask for its name. The expected answer is Cedar because the information is in the context. Then open a new chat and observe the result, checking the memory settings: neither remembering nor failing to remember demonstrates a parameter update. As a variation, provide a corrected name, Alder, in the same chat: check that the correction takes precedence. Reuse this test to distinguish conversational behavior from product memory; do not use it as a complete audit of data retention.

Are They Objective and Neutral

They reflect human biases.

Because they are trained on data from the Internet, they inherit the same biases, stereotypes, and assumptions present in human language. If not managed properly, these elements can emerge subtly in responses.

Are They One Hundred Percent Safe or Reliable

No AI system is perfect.

LLMs can hallucinate, give wrong advice, or respond unpredictably to certain prompts. They should always be used with human supervision, especially in critical fields such as healthcare, law, or education.

What Is the Lesson

When choosing a use for an LLM, ask what result you need and how you will verify it. A fluent answer may contain errors, biases, or proposals unsuitable for the context. The model's capabilities become useful when you have relevant data, a check of the result, and criteria for deciding whether to accept it.

When Overconfidence Meets Artificial Intelligence

How to recognize uses in which a plausible answer is accepted without checking. The following twelve stories preserve the problems in the guide: corroborated cases are distinguished from illustrative scenarios.

In the stories that follow, the decisive point is the omitted check: a source not sought, a confidentiality condition ignored, or a result not reread. Identifying that step lets us apply the lesson to another case without having to repeat the same events.

Stories labeled "illustrative scenario" present situations to reason about without attesting to identified events. They help us recognize a risk or a choice; they do not demonstrate that an episode actually happened. Documented cases, by contrast, are accompanied by sources supporting the reported details.

1. The Lawyer Who Cited Invented Cases

In the 2023 Mata v. Avianca proceedings, lawyers submitted nonexistent precedents obtained through ChatGPT. The order of June 22, 2023 documents six invented decisions and a $5,000 sanction against the lawyers involved and their firm. The episode does not demonstrate that using an LLM is unlawful in itself: it shows the problem of submitting unverified material and insisting after its existence has been challenged.

> Essential reference — A documented case. Judge P. Kevin Castel's order is a primary source: it describes the filings, checks, and sanctioned conduct. We do not attribute the overall outcome of the case to AI alone. The transferable step is to check every citation in the authentic document before using it.

https://www.nhd.uscourts.gov/sites/default/files/pdf/Mata-v-Avianca-sanctions-order.PDF

2. The Chief Executive Officer Who Disclosed Their Secrets

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

A financial services executive used an online chatbot to “improve the tone” of a confidential report before a board meeting. In the scenario, the executive sends confidential material to an external service without checking its terms. We then imagine that information is reused or exposed: this is a risk to analyze, not an automatic effect of the cloud or proof that every input is learned and returned to other users.

> Moral: before requesting a text revision, check whether you may send it to the chosen service. An improvement in tone does not authorize sharing confidential information.

3. The Student Who Outsourced Thinking

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

A university student in Lyon submitted a flawless essay on the ethics of technology, so flawless that it included the sentence “As an AI language model, I cannot form opinions.”

The student had not even reread the output. The professor did not fail the student for using AI, but for not reading what was being submitted.

> Moral: using the model for a draft does not remove the work of reading, understanding, and checking what you submit.

4. The Recruiter and the Algorithmic Ghost

Reuters reported in 2018 that Amazon had abandoned an experimental résumé screening system after detecting penalties associated with gender. The system had been built using historical data in which male candidates predominated. The report does not support the cancellation of dozens of offers or the public apologies described in the guide: those details are not presented again as facts.

This case concerns a machine learning recruitment system, not an LLM chatbot. We retain it because it teaches us to check a common problem: a prediction can reproduce distortions in data. To verify this, we need to compare equivalent cases and results, without assuming that a neutral description of the tool guarantees fair decisions.

(Journalistic source: Jeffrey Dastin, Reuters, October 10, 2018, Amazon scraps secret AI recruiting tool that showed bias against women. The report also relies on anonymous sources; it is not a technical publication about the system.)

https://finance.yahoo.com/news/amazon-scraps-secret-ai-recruiting-174002175.html

5. The Influencer Who Asked AI for Medical Advice

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

A lifestyle influencer in Seoul began offering “AI wellness tips” generated by a chatbot. A follower took a combination of supplements suggested by AI and ended up in the hospital. Nobody knew whom to hold responsible: the influencer, the model, or the audience that had believed the advice.

> Moral: when health is at stake, “sounding confident” does not mean “being right.”

6. The Programmer Who Let the Model Write the Code

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

A junior developer proudly used an LLM to generate a script intended for production. The developer copied it directly into the company's cloud pipeline, including a treacherous loop that deleted operational data during an update.

Recovery took two days. The model did not lose its job. The developer did.

> Moral: before running code on operational data, read what it does and test its behavior in a controlled environment. Fast generation does not establish program safety.

7. The Politician and the AI Speech

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

In 2024, a local politician in Eastern Europe delivered a passionate speech about freedom and democracy, which was later discovered to have been written entirely by AI. Unfortunately, the text had been taken almost word for word from a Soviet propaganda manual from the 1980s, rephrased by the model.

> Moral: even a model capable of recognizing irony can produce inappropriate text. Check content and context before using it.

8. The Couple Who Asked AI for Legal Advice on Divorce

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

To save money, a couple in the United Kingdom used an LLM to generate their divorce agreement. The document looked official, until the court rejected it

because it contained an impossible clause about “custody of the family Wi-Fi password.”

> Moral: a document’s formal appearance does not establish that its clauses are valid. Using it in an actual legal situation requires a competent review of the case and applicable rules.

9. The Journalist Who Checked Nothing

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

A small news outlet published an article about a “revolutionary mission to Mars,” citing detailed statements from NASA scientists. The story went viral. NASA had never heard of either the mission or the scientists.

The quotations were AI inventions. The newspaper published an apology; the editor lost their job.

> Moral: before publication, check statements and names against authentic sources. A quotation’s apparent precision does not establish that it exists.

10. The Engineer and the Data Leak

In May 2023, restrictions on Samsung's use of generative tools were reported after confidential code was sent to an external service. The problem documented by the reports is the transmission of sensitive information. This is not enough to claim that the code was incorporated into parameters or made available to other users.

> Example — Before pasting code. Check whether you may transmit it, which service receives the data, where it is stored, and for what purposes. If you do not have these answers, use fictional data or an authorized environment. Replacing variable names alone can leave confidential logic recognizable. A local model can reduce transmission to third parties, but logs, access, and backups also need to be checked.

(Corroboration: Thomson Reuters Institute, ChatGPT and Generative AI within Corporate Law Departments, 2023, which reports Samsung's restrictions. It is an industry report, not an audit of Samsung's systems.)

https://www.thomsonreuters.com/en-us/posts/wp-content/uploads/sites/20/2023/05/ChatGPT-Generative-AI-in-Corporate-Law-Departments-2023.pdf

11. The Judge Who Consulted ChatGPT

In judgment T-323 of 2024, Colombia's Constitutional Court examined the use of ChatGPT to provide reasons for a decision issued on January 30, 2023. The case concerned a minor's access to healthcare services.

> Essential reference — Responsibility for judgment. The judgment is a primary source for reconstructing those proceedings. For the reader, the lesson is to distinguish consulting a tool from responsibility for the decision: answers must be checked against the specific case. The Court considered the decision valid: the judge had first performed his own analysis; it also examined limitations in the application of the principles of transparency and accountability. We do not derive a universal legal rule from a Colombian judgment.

https://www.corteconstitucional.gov.co/relatoria/2024/T-323-24.htm

12. The Teacher Who Let AI Grade Essays

> Illustrative scenario — The characters, places, and outcomes in this story are not presented as a verified news case. They serve to discuss a concrete risk.

An American teacher used a language model to automatically grade a stack of student essays. It worked perfectly until a student challenged a failing grade and the model, reassessing the essay, judged it “excellent.”

The teacher had to grade all the papers again by hand.

> Moral: an automated assessment needs comparison with explicit criteria and consistent results. Whoever assigns the grade must be able to explain the decision and handle a challenge.

A Final Reflection

All these stories share the same plot: people trusted language more than logic. In documented cases and illustrative scenarios, the risk arises from accepting output without adequate checking. The system's abilities do not eliminate the responsibility of those who choose to use it.

For each use, identify where a proposal becomes a decision: submitting an essay, publishing an article, or running a program. The relevant check should come before that step. The model can help prepare it, but responsibility for the choice remains with whoever uses the result.

Part IV — Applications and Examples

Real-World Examples

You may not see them, but LLMs are already working behind the scenes in many fields, helping people save time, make decisions, and access information.

Here are some concrete examples of how they are used today.

Healthcare

Doctors and nurses spend substantial time on documentation. An application using an LLM can draft a clinical note from a transcript, summarize a patient's history, or simplify an explanation. Whoever uses the result must compare it with the original information: an omission or an added detail can change the meaning of the note.

> Example: a doctor records a note about a patient's condition, and the LLM turns it into a clear, structured report for the medical record.

Education

In schools and universities, LLMs support both students and teachers: they explain difficult concepts in simple language, generate practice or quiz questions, and provide feedback on students' writing.

> Example: a high school student asks "Can you explain how photosynthesis works?" The LLM provides a simple, age-appropriate explanation.

Journalism and Writing

Writers use LLMs to find headline ideas, suggest alternative wording, and summarize interviews or research.

> Example: a journalist pastes interview notes and receives a first draft of an article in return.

Customer Service

Companies use LLM-based chatbots to answer frequently asked questions, help people navigate a website, or guide a purchase. To check an answer, compare it with the service's procedure and confirm that its conditions, prices, and steps are the actual ones. Time saved should be measured with corrections and any staff intervention included.

> Example: a customer writes "I cannot reset my password," and the LLM guides them through the necessary steps.

Business Activities and Productivity

LLMs help professionals in every field generate reports or meeting summaries, translate business documents, and write emails or proposals more quickly.

> Example: a project manager asks the LLM to “summarize last week's meeting notes and prepare an action plan.”

Creativity and Art

Artists and creators use LLMs as a source of inspiration: to write poems or scripts, generate dialogue between characters or story ideas, and collaborate with musicians and designers.

> Example: a screenwriter stuck on a scene asks the LLM to suggest three ways a character might get out of a difficult situation.

Everyday Tools Quietly Powered by AI

Many of the tools you use every day, from writing assistants to intelligent search engines, are quietly powered by LLMs. You may not see AI at work, but it is there: helping behind the scenes.

The use of these models can change many everyday activities. Their useful contribution depends on the task: a draft that can be corrected may save time, while an answer that is hard to check may add work. Effects on people and professional roles also depend on how the organization chooses to use the tool.

Opportunities and Benefits — Why LLMs Can Be Truly Useful

Despite their complexity, LLMs are not just for scientists or technology companies. Some of their greatest benefits lie precisely in how they help everyone: from busy professionals to students, from writers to customer service teams.

Let us take a moment to consider the opportunities and benefits these models offer when used carefully.

Saving Time on Repetitive Tasks

One of the most immediate benefits is the amount of time these tools can save. Instead of spending an hour drafting an email, summarizing a document, or writing meeting notes, you can ask the model to prepare a first draft. Then you review and adapt it.

A quickly generated draft may free time for other activities. To find out whether it actually does, include the time needed to read, check, and correct the result. A useful assistant lets you complete the work more effectively; generation speed alone does not demonstrate that benefit.

Making Knowledge More Accessible

Not everyone learns in the same way. Some people prefer visual explanations; others need simplification. LLMs can adapt the tone and complexity of information to each user's needs.

They can break down difficult topics, translate jargon into everyday language, or explain a concept like a friendly tutor. Whether you are fifteen or fifty-five, an LLM can help you understand things at your own pace.

Helping with Language and Communication

LLMs are extremely useful for people who struggle to write or express themselves, especially those who are not using their native language, people with different ways of learning, or those working across cultures and languages.

Need to write professionally but are unsure about the tone? The model can help. Working in English but thinking in Italian? The model can bridge that gap.

Sparking Creativity

Creativity does not always come on command. Sometimes all it takes to get started is a small spark: a new expression, a different viewpoint, or a bold idea we had not considered.

LLMs can act as brainstorming companions, expanding your imagination rather than replacing it. They can suggest new viewpoints for a story, titles for a report, or different ways of framing a presentation.

They are not artists, but they can help you feel like one.

Assisting Without Replacing

You can use an LLM to take notes, propose alternatives, or prepare a check. The image of a digital copilot helps describe this support, provided that responsibility for decisions is clear: a person must set the task, accept or correct the result, and take responsibility for the next action.

The person remains in control, but receives powerful help. Used responsibly, LLMs can make us more productive, more expressive, and more confident in everyday work.

> Guided activity — Measuring a real benefit. You need to extract actions from a fictional meeting note: “Luca sends the quotation on Friday; Sara checks the amounts on Monday.” First extract the two actions by hand, then compare them with the model's output and record the time required, including verification. As a variation, add “Perhaps we will request a third quotation”: this must not automatically become a definite assignment. Reuse the tool if quality and total time improve; if correcting costs more than performing the task, choose another procedure.

Unexpected Stories from the World of Language Models

Strange, Surprising, and Sometimes Disturbing Stories

LLMs can have effects that go beyond text. The following cases require attention to source quality and the difference between a news account and a causal explanation.

The documented cases show different consequences: a text may entertain, support an activity, or lead someone to trust incorrect information. Read each case by distinguishing what the source documents from explanations offered to interpret it. The fact that an error is plausible does not by itself establish how it was produced.

The more we understand how LLMs behave in the real world, the better we can use them responsibly, with wonder and caution.

A Chatbot, a Tragedy, and the Importance of Boundaries[2]

In March 2023, La Libre reported the death by suicide of a man after weeks of conversations with the chatbot Eliza in the Chai app. The family's account raised questions about the system's responses and protections in contexts of vulnerability. A news account of this kind does not, by itself, establish a single causal link; we avoid reducing a complex event to an automatic explanation.

The example points to a concrete limitation: a general-purpose chatbot is not a substitute for professional assistance. Designing a service for emotional contexts requires expertise, management of critical situations, and evaluation of responses, not just a reassuring style.

https://www.lalibre.be/belgique/societe/2023/03/28/sans-ces-conversations-avec-le-chatbot-eliza-mon-mari-serait-toujours-la-LVSLWPC5WRDX7J2RCHNWPDST24/

An AI-Assisted Story in a Literary Competition[3]

A report from March 2016 describes the submission of computer-assisted stories to the Hoshi Shinichi award and one work passing the first selection round. The 2021 date given in the guide does not correspond to this episode. Passing an initial stage is not equivalent to being a finalist or winner; the project involved human contributions and should not be presented as proof of an autonomous artificial author.

2 La Libre, March 28, 2023, “Sans ces conversations avec le chatbot Eliza, mon mari serait toujours là” (“Without these conversations with the chatbot Eliza, my husband would still be here”). A journalistic account also based on the family's testimony; the article is not a clinical study. Access to the full text may be limited.

3 Dentsu-ho, AI Novel Creation Project Reports on Submission to Nikkei’s Seiichi Hoshi Award, March 25, 2016. It is an account of the project and selection, not a demonstration of the abilities of current LLMs.

The case allows us to discuss authorship, originality, and human contribution. To evaluate it, ask which part was chosen, written, or revised by people and which by the system. Without this information, the headline “story written by AI” alone does not adequately describe the work.

https://dentsu-ho.com/en/articles/3855

Part V — From General to Specialized Intelligence

From General to Specialized — How LLMs Learn New Skills

Most LLMs start out as generalists. They are trained on enormous amounts of text from books, articles, and websites, ranging from Shakespeare to weather reports and Wikipedia. This makes them very good at a broad range of tasks, but not perfect for specific needs.

Think of a general LLM as someone who has read a million books. Impressive, certainly, but perhaps not the best person to ask immediately for legal advice or to write a scientific paper.

How do we make a model more specialized? This is where a process called fine-tuning comes in.

What Fine-Tuning Is

Fine-tuning is further training of an existing model using examples selected for a field or task. During this phase, the parameters—the numerical values the model uses to produce responses—change. Putting examples into an ordinary request instead provides context for the response without updating those parameters.

For example:

- A hospital might fine-tune a model on medical records to help doctors write reports.
- A company might use customer service transcripts to improve chatbot responses.
- A school might use teaching materials to help explain science or history topics.

The base model has already acquired language patterns during training. Fine-tuning uses more examples to steer its responses toward an objective, such as a more consistent format or a particular task. Comparing it with a course for a new job describes adaptation, but does not give the model experience or a professional qualification.

Why It Matters

Fine-tuned models can be more accurate in their domain, use the right tone, vocabulary, and style, and avoid common errors that a generic model might make.

For users, this often means better results, fewer revisions, and greater confidence in the output.

A Real-World Analogy

Imagine hiring someone who has read every book in the library. Now you need them to help customers at a technical support desk. You would probably spend a few days training them on your products, communication tone, and procedures.

To find out whether fine-tuning worked, evaluate the model on examples you did not use to train it. Compare errors before and after adaptation, including tasks the model could already perform. Improvement on the training data is insufficient: the model may reproduce them well and fail on new cases, or perform worse on previous tasks.

> Guided activity — Choosing between examples, retrieval, and training. You want answers in the format “Question / Answer.” A generic question does not define the style. Show two fictional examples and check that a third answer respects the format: you are using context without updating parameters. As a variation, ask about a company rule that has just changed: style examples do not provide that fact, so an up-to-date document must be retrieved. Reuse examples to guide the format, retrieval to provide sources, and fine-tuning when you have data and an evaluation that justify its cost.

LLMs as Memory Machines — Retrieving Knowledge and Human Experience

An LLM can help you explore ideas and knowledge represented in its training texts, as well as write emails or summarize documents. The metaphor of a living archive expresses this possibility, but differs from a searchable archive: the model generates an answer and may combine, alter, or invent information. When you need a precise reference, you must find and check the source.

Training collections may include books, dialogue, history, science, philosophy, poetry, and everyday conversations. The model learns patterns from these examples. The choice of collections determines which voices and topics it encounters: this does not make any model a representation of all human thought.

Preserving What We Know

LLMs can help us rediscover forgotten knowledge. They can bring out ideas from less well-known texts, make complex theories more accessible, and bridge generations.

Want to discover how letters were written in the eighteenth century? Curious about ancient philosophical ideas? Looking for how a problem was solved in a 1996 research paper?

The model can reproduce and reorganize learned information without remembering it as a person remembers an experience. To use an explanation today, compare it with a suitable, current source. If you request a quotation, separately check that the text exists, the wording is faithful, and the attribution is correct.

Amplifying Human Experience

Texts express emotions, relationships, and lived experiences: hardship, joy, doubt, learning, and identity. A model trained on varied texts can reproduce these ways of expressing them. A conversation may therefore resemble an exchange with many voices, but that impression does not show that the system represents every culture or has a collective mind.

The model can produce words about suffering, beauty, or hope using relationships learned from texts. A sensitive response does not demonstrate that it has those experiences. Assess whether the words suit the situation and help the person reading them.

In this sense, LLMs are not just productivity tools. They are cultural tools, capable of bringing out stories, values, and perspectives at risk of being lost in the noise of the present.

Knowledge Rediscovered and Reimagined

Educators can use LLMs to bring distant topics closer to their students. Researchers can find unexpected connections between disciplines. Writers and artists can draw inspiration from voices from a remote past.

And individuals, curious and reflective people, can explore new ideas or return to old ones with a guide that never gets tired of the request “Tell me more.”

> Guided activity — Retrieval is not enough. You need to answer a question about the duration of a fictional warranty. Give the system two documents: version 1, duration twelve months; version 2, effective October 1, duration twenty-four months. Ask for the duration, version, and passage used. For a November request, the expected result is twenty-four months with a reference to version 2. Verify both retrieval and the answer. As a variation, remove the valid version: the system must state the limitation without inventing. Reuse retrieval if sources, access, and versions are controlled; outdated documents can worsen the answer.

A Whole World of Language Models — From Giants to Little Helpers

When people hear about a “language model,” they often think of something like ChatGPT or Google Gemini: enormous, powerful systems that seem to know everything and be able to do almost anything. But the reality is much richer and more varied.

Today, language models come in different shapes and sizes, each designed for different needs and situations.

We now live in a world where language models range from gigantic cloud-based systems capable of answering complex questions to small models that run on laptops or even small peripheral devices. Let us explore this landscape.

The Giants — Cloud-Based Generalists

The guide mentions families such as GPT, Gemini, Claude, and LLaMA. These names are examples of the model landscape, not a list of versions currently available. Model size and execution location are separate characteristics: a large model can also run on your own infrastructure.

These systems are trained on enormous datasets and have billions, sometimes hundreds of billions, of parameters. They can write essays, summarize reports, explain scientific topics, generate code, and even hold conversations that seem surprisingly natural.

They are extremely capable, but also require a great deal of computing power. This is why they usually reside in the cloud and are used through web platforms or APIs, interfaces through which other programs can access the service.

For a remote service, check quality on your task, which data you may send, and what information the provider makes available about the model. Also consider costs and dependence on a connection. Difficulty reconstructing internal processing concerns the model and does not automatically disappear when you run it on your own device.

The Specialists — Medium-Sized Fine-Tuned Models

Between very large and small models are medium-sized models. Size and execution location are separate choices: a model does not become local merely because it is small. Some release their weights, the model's numerical parameters. Before adapting or distributing them, check the license: having the weights does not automatically mean having an open-source license. Finally, measure whether the available hardware provides suitable speed and quality on your task.

Think of models such as Mistral, Mixtral, Falcon, and Dolly. Some are optimized to run on standard GPUs, graphics processors, in a laboratory or office. Others are adapted to medical, legal, or customer service domains.

In business, education, and research, a model of this size may be an alternative to evaluate. The balance between resources and quality must be demonstrated: test examples from your work and measure memory requirements, time, and errors. Whether it can run locally depends on the available hardware.

They are often chosen when you want to retain control over your data and adapt the model to your specific context.

The Smallest — Lightweight Models for Local Use

At the other end, we find small, highly efficient language models capable of running on laptops, Raspberry Pi boards, or even edge computing devices, which process data close to where it is produced.

These models might have only a few hundred million or a couple of billion parameters, but they are surprisingly capable.

Examples cited include TinyLLaMA, GPT2-small, and DistilGPT, as well as compressed or quantized models. Quantization represents numerical values with reduced precision to lower memory requirements; its effect on quality must be measured. A model can work offline if the required files and software are already on the device and the application does not require external services.

Local execution can be useful when a network is unavailable or when you want to keep data on the device. To achieve that result, also check what the application does: logs, backups, and connected services may send data elsewhere. Then assess whether the device completes the task with acceptable quality and speed.

They can generate texts and explanations even on complex topics; quality and manageable length depend on the model and resources. Evaluate whether they are sufficient for the task rather than infer an absolute inability from size alone.

Choosing the Right Solution

There is no model suitable for every situation. What matters is matching the tool to the task.

For a creative writing assistant, compare cloud and local models on the same passage and evaluate the suggestions. For a tool intended for field technicians, test operation without a network. For an application handling health data or other confidential information, check where the data travel and who can access them. For a specialist task used by lawyers, doctors, or teachers, compare the base and adapted models on new, checkable examples. The results of these tests justify the choice.

When choosing, consider three questions separately: does the model perform the task well? Can you run it where needed, with the available hardware and connection? Do you have the rights and conditions required to use the model and data? One label, such as “large” or “local,” does not answer all three.

Beyond Words — Interpreting Images Alongside Text

Until recently, most language models could do only one thing: work with text. You typed a question and received an answer in words.

Now a new generation of models is learning to do more. Much more.

A multimodal model processes more than one type of data. For example, it may receive an image along with a written question, or audio to transcribe. Check which input and output types the particular model supports: “multimodal” does not mean that every system can read, listen to, and generate any kind of content.

What This Means in Practice

Imagine these situations:

- You upload a photo of a chart and ask “What is happening here?” The model explains the trend in simple terms.
- You send an image of your handwritten math homework. The model reads it and guides you through the solution.

- You paste a page from a scanned PDF and ask for a summary. The model reads the text and interprets the tables and layout.

This is no longer science fiction. These possibilities already exist in tools such as GPT-4V, Google Gemini, and other advanced AI systems.

Why It Is So Important

Because in real life we do not use only words. We combine language with images, sounds, and symbols. A doctor looks at diagnostic images, a teacher explains with drawings, a mechanic reads dashboards and sensor charts.

Combining text and images lets you ask questions that text alone cannot support, such as reading a label in a photograph. Usefulness is checked by comparing the answer with what is visible. Combining different inputs does not, by itself, show that the model understands the world as a person does.

They can help students learn with diagrams and step-by-step explanations, researchers analyze visual data or technical figures, doctors receive AI assistance in understanding images, and people with disabilities interact with AI through voice, gestures, or visual cues.

What Comes Next

We are only at the beginning. These new models can already describe the contents of an image, explain a chart, or answer questions about a photograph. Processing video requires connecting successive images and following their changes; combining voice and images also requires appropriate multimodal support. Check the actual abilities of the version being used, without assuming that all models support these inputs.

The check must distinguish what the image shows from what the model infers. If text is unreadable or an object is hidden, a precise answer may still be a guess. Ask the system to flag these limits and compare the result with the image before using it.

One thing is clear, nevertheless: the future of AI is not just about better words. It is about connecting words with the world around us.

> Guided activity — A photo is not the whole world. You want to count objects: the question “What is happening?” encourages interpretations. Draw three clearly separated circles and ask only for the number of visible circles: the expected result is three. Check the image directly. In the variation, partially overlap the shapes and ask the model to distinguish what is visible from what is hypothesized. Reuse vision for tasks with corroboration; an ambiguous image does not authorize conclusions about intentions, identities, or facts outside the frame.

Part VI — Infrastructure and Sustainability

Power and Price — The Hidden Cost of Intelligence

LLMs can do impressive things: write essays, explain complex topics, and even simulate human conversations. But behind the scenes, these abilities have real costs.

Running and training these models is not just about clever code. It requires enormous amounts of data, energy, and computing power. Let us see what this entails.

The Cost of Training

Training a state-of-the-art model such as GPT-4 or Gemini requires:

- Billions of words from books, articles, and websites.
- Weeks or months of computation on thousands of high-end GPUs.
- Millions of dollars for hardware, storage, and energy.

Comparing two training runs requires separate measurements. Financial cost states how much was spent, for example in euros. Energy consumption may be measured in kilowatt-hours; associated emissions also depend on how that energy was produced. Hardware, duration, and machine utilization affect the result. Comparing a run with “a small city for days,” without defined consumption and a time period, does not allow the calculation to be checked.

These costs make large models accessible mainly to technology giants or governments, unless we seek alternatives.

Running the Model — Inference Costs

After training, producing responses requires further computation. This phase is called inference: the model uses its existing parameters to process the input and generate the output. An individual request is not a new training run, but consumes resources and incurs costs:

- Cloud models charge per token or per request.
- Response time depends on the model, hardware, load, text length, and tools used.
- Energy and emissions also depend on system efficiency and electricity sources; size alone is not enough to calculate them.

If you run a customer service chatbot or an educational application, inference performance and cost quickly become important.

What About Sustainability

With so much energy involved, concern about AI's environmental impact is growing. Data centers require cooling systems, continuous electricity, and water; the carbon footprint of a single large model can be significant.

To reduce resource requirements, developers explore smaller models such as TinyLLaMA, DistilGPT, and Mistral, and compression techniques. Quantization lowers the precision used to represent numerical values, reducing memory requirements. Execution on the device may avoid cloud transfers. None of these choices alone guarantees energy savings: measure them alongside quality and the time needed for the task.

These approaches can reduce the resources required and increase control over the environment. The benefit must be measured: a local system still requires access management, updates, and data protection.

Choosing the Right Balance

Bigger does not always mean better. For many tasks, such as summarizing emails, preparing reports, or guiding users, a medium-sized or optimized model can work well without high costs.

First define an acceptable result, then measure the resources needed to obtain it. Compare models on the same task and include correction work. A fast system requiring many corrections may be less useful than a slower system that meets the requirements.

Comparing AI Models — Cloud Giants and Local Experts

The choice starts with requirements: what result do you need, by when, using which data, and at what cost? Test alternatives on the same set of cases. A large cloud model may help with a difficult task; a smaller local model may suffice for a narrowly defined task. Location and size alone establish neither quality nor confidentiality.

Here is a simple comparison to understand benefits and trade-offs.

Feature	Large cloud-based LLM, for example GPT-4 or Gemini	Optimized local model, for example Mistral or TinyLLaMA
Performance	Must be measured on the task: depends on version, tools, and input.	Must be measured on the task; adaptation can help, without guaranteeing superiority.
Access	Requires Internet, an API, or a platform.	Can work offline on a laptop, server, or edge device.
Speed	Depends on servers, load, connection, and text length.	Depends on hardware, model, load, and text length.
Cost	Check prices, subscriptions, and request volume.	Hardware, energy, maintenance, and management; compare total cost.

Feature	Large cloud-based LLM, for example GPT-4 or Gemini	Optimized local model, for example Mistral or TinyLLaMA
Energy consumption	Depends on infrastructure, utilization, and load; requires comparable measurements.	Depends on hardware and utilization; local deployment does not guarantee savings.
Privacy	Transmission to a provider: check contract, access, and retention.	Can avoid external transmission; still check access, logs, and backups.
Customization	Depends on the options and constraints offered by the service.	Depends on the license, access to weights, tools, and available resources.
Tasks to test against requirements	Complex questions, creative writing, and research: compare quality, sources, and costs.	Customer support, education, and connected devices, known as IoT: check quality, resources, and operation where needed.

Guided activity — Evaluating cost and constraints together. For one hundred requests, a fictional service costs 0.02 euros per request: the variable cost is 2 euros. A local alternative costs 20 euros to manage over the period, plus any hardware and energy costs. The price per token alone does not settle the choice. Check the calculation and also measure errors and time. As a variation, increase requests to two thousand: the service costs 40 euros, but you need to check whether the alternative can handle that load. Reuse the comparison for the measured volume and the same requirements; do not automatically transfer prices or performance to another environment.

Part VII — Appendices

Before You Ask — A Checklist for Using LLMs Wisely

Before using a result, clarify the task, provide the necessary data, and decide how to check it. The following list helps you prepare these steps. Apply it to a concrete piece of work: an answer that is pleasant to read is not yet an answer ready to use.

Here is a quick checklist for using LLMs responsibly, effectively, and safely.

Area	What to do	What to avoid
Know the tool	Understand the model you use: name, size, and purpose.	Using it blindly without knowing how it works.
Formulate clear prompts	Be specific, provide context, and define roles, for example “You are a teacher…”.	Writing vague or ambiguous prompts.
Think critically	Always double-check facts, especially in sensitive fields.	Trusting the output without verifying it.
Pay attention to bias	Test the model in different scenarios and from different perspectives.	Assuming the model is neutral or fair by default.
Protect privacy	Avoid pasting personal or confidential data into public models.	Assuming the system is private or secure by default.
Use responsibly	Consider the impact of the output and the people involved.	Using LLMs to automate tasks without human supervision.
Refine when necessary	Revise prompts and instructions iteratively to improve results.	Expecting perfect answers on the first attempt.

Conclusion

LLMs are changing how we interact with technology. They can write, translate, summarize, explain, and even surprise us with their creativity.

An LLM produces tokens using relationships learned during training and the context supplied with the request. This process can generate useful texts and solutions to complex tasks. To accept an answer, however, compare it with what the task requires: a source, a calculation, a checked example, or a quality criterion. Fluent wording does not demonstrate correctness, emotions, or subjective experience.

Yet, when used carefully, they can be powerful partners:

- They help us save time.
- They make complex ideas more accessible.
- They support learning, creativity, and communication.

Control requires specific actions: setting the task, deciding which data to provide, checking the result, and choosing who may use it to act. In this way, an LLM can

support work and suggest ideas without automatically receiving authority to make decisions. Its usefulness depends on how you organize these steps.

Used with care, curiosity, and responsibility, these models can help us do more and think better.

To test what you have learned, choose an activity different from the worked examples, such as summarizing a school notice or extracting a deadline from a work note. Write down the expected result, prepare the request, and compare the response with the document. Then remove a necessary detail and check how the result changes. Knowing when to stop and seek missing information is part of the skill, alongside knowing how to formulate the request.

Glossary

Term	Definition
AI — Artificial Intelligence	Methods through which a system performs tasks such as classifying data, recognizing patterns, or producing responses. The name does not imply that the system thinks or perceives like a person.
Language model	A system trained to predict and generate human language on the basis of patterns in text.
LLM — Large Language Model	A model trained on large text collections to process and generate language. Its performance must be checked on the task; a readable answer does not establish correctness.
Prompt	A natural language instruction or question given to the model to guide its output.
Prompt engineering	The practice of designing clear, effective prompts to obtain better results from LLMs.
Token	A unit into which text is divided: it may be a word, part of a word, or a punctuation mark. During generation, the model progressively selects further tokens.
Inference	The process by which a trained model generates a response.
Fine-tuning	Further training of an LLM on specialized data to adapt it to specific tasks or fields.
Bias	Unintentional assumptions or stereotypes reflected in the model because of patterns in training data.
Hallucination	A situation in which an LLM generates false but convincing information.
Emergent Capability	Performance observed in larger models and not detected in smaller ones in the same evaluation. The result also depends on the measure chosen; it does not establish a new general faculty.
Multimodal Model	A model that processes more than one data type, such as text and images. Available input and output types depend on the particular system.
Cloud-based model	An LLM used through the Internet, running on remote servers.

Term	Definition
Local model	A model running in an environment controlled by the user; possible sizes depend on available hardware.
RAG — Retrieval-Augmented Generation	A system searches for relevant documents and adds passages to the model's context. The model uses that context to answer; both the retrieved documents and the answer's faithfulness must be checked.
Quantization	Representing numerical values with reduced precision to lower memory requirements. It can make execution on a device easier, but its effect on quality must be measured.

Want to Know More — Resources to Explore

Foundational Reading

Attention Is All You Need

Vaswani and colleagues, 2017. The work introduces the transformer and evaluates its performance in translation. This architecture underlies many modern LLMs; the original work does not demonstrate the abilities of every subsequent model.

https://arxiv.org/abs/1706.03762

Man Is to Computer Programmer as Woman Is to Homemaker

Bolukbasi and colleagues, 2016. A study of gender bias in word embeddings, numerical representations of words, which also proposes a method to reduce some of its manifestations. It is not proof that an intervention eliminates every form of bias.

https://arxiv.org/abs/1607.06520

On the Dangers of Stochastic Parrots

Bender and colleagues, 2021. A critical analysis of risks, biases, and ethics in large language models.

https://dl.acm.org/doi/10.1145/3442188.3445922

European Union — Artificial Intelligence Regulation

Regulation (EU) 2024/1689 is available on EUR-Lex, the official portal for European Union law. For a concrete use, consult the current text, applicable scope, and deadlines; this guide does not replace a legal assessment. An informational website about the AI Act is not the European Commission's website.

https://eur-lex.europa.eu/eli/reg/2024/1689/oj

Educational Guides

OpenAI — Prompt Engineering Guide

Practical examples and suggestions for constructing better prompts.

https://platform.openai.com/docs/guides/prompt-engineering

DeepLearning.AI — ChatGPT Prompt Engineering for Developers

A short course with examples for developers. Check the website for access arrangements and prerequisites, which may change.

https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

OECD — AI and the Future of Skills

Reports on AI's impact on education and work.

https://www.oecd.org/en/topics/artificial-intelligence-and-education-and-skills.html

References for Further Reading

These references support the explanations and distinctions developed in the guide. The examples with receipts, pencils, prices, and company documents are exercises with fictional data. The sources do not certify those data or the performance of a specific product.

Emergence and Evaluation

Wei and colleagues, Emergent Abilities of Large Language Models, 2022. On the relationship between scale and observed performance.

https://arxiv.org/abs/2206.07682

Schaeffer, Miranda, and Koyejo, Are Emergent Abilities of Large Language Models a Mirage?, 2023. On the role of the metric in apparent performance jumps.

https://arxiv.org/abs/2304.15004

Explanations and Their Limitations

Turpin and colleagues, Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting, NeurIPS 2023. On the distinction between a plausible explanation and a faithful account of influences on the result.

https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract.html

Information Retrieval

Lewis and colleagues, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 2020. The work combines a generative model with document retrieval and evaluates tasks requiring knowledge. It does not guarantee that every retrieved source is current or correct.

https://arxiv.org/abs/2005.11401

Memory and Data Handling

OpenAI, documentation on ChatGPT memories. Describes product features: it should not be generalized to every LLM.

https://learn.chatgpt.com/docs/customization/memories

OpenAI, Data controls in the OpenAI platform. Distinguishes training use from data retention in APIs. Terms may change; recheck them for the applicable service and contract.

https://developers.openai.com/api/docs/guides/your-data

How to Read News Cases

Sources for the Mata order, the Colombian ruling, the Thomson Reuters Institute report, and the Amazon, Eliza, and Hoshi accounts appear beside the cases. News accounts have a different evidential value from court records and experimental studies. Full access to some articles is restricted: we must distinguish what the accessible document supports from details we cannot verify. A news report can help reconstruct an episode, but does not replace a court document or demonstrate an experimental result.