

Large Language Models (LLMs): How ChatGPT-Style AI Reads, Writes and Gets Things Wrong

What a large language model is, how it writes one token at a time, how it is trained, how to prompt it well, why it hallucinates, how to choose one, and what happens to the data you type in.

20 min read · Content last verified 8 October 2026 · Reviewed by Shravya M

Free awareness guide. Not legal or financial advice. <https://www.shravyadigital.com/guides/large-language-models>

CONTENTS

1. Before you start
2. Chapter 1: What an LLM is
3. Chapter 2: Tokens and the context window
4. Chapter 3: How LLMs are trained
5. Chapter 4: Asking good questions (prompting)
6. Chapter 5: Why LLMs make things up
7. Chapter 6: Choosing an LLM
8. Chapter 7: LLMs and your data
9. Key words

Before you start

This guide is the reading for **Course 2** of the AI Foundations Pathway. It takes about 20 minutes. It builds on Course 1, *How AI Models Work*.

You'll learn what a large language model (LLM) is, how it produces an answer one piece at a time, how it is trained, how to ask it good questions, and why it can be confidently wrong.

All names and examples here are made up.

Chapter 1: What an LLM is

A **large language model (LLM)** is a generative AI model trained on an enormous amount of text (books, websites, articles, code) to do one thing extremely well: **predict the next piece of text**.

Give it "*The capital of India is*" and it predicts "*New Delhi*". Do that thousands of times in a row, each time adding the new piece and predicting again, and you get paragraphs, emails, summaries, translations and code.

It's called **large** because of its size: billions of parameters, trained on more text than a person could read in many lifetimes.

Chatbots like ChatGPT, Gemini and Claude are apps built on top of LLMs. For a comparison of the main ones, see our guide [Popular AI Chatbots and LLMs Compared](#).

The rule: An LLM predicts likely text. Likely and true are usually the same, but not always.

Chapter 2: Tokens and the context window

LLMs don't read whole words. They read **tokens**: chunks of text. In English a token is roughly $\frac{3}{4}$ of a word; "unbelievable" might be three tokens. Hindi and other Indian scripts usually need **more** tokens for the same sentence, which can make them slower and costlier to process.

The **context window** is how many tokens the model can consider at once: your question, any documents you paste in, the conversation so far, and its own answer. Modern models range from tens of thousands to over a million tokens.

What this means for you:

- In a very long chat, early details can fall out of the window or get less attention. Restate what matters.
- Paid services usually **charge by the token**, both what you send and what it writes.

Chapter 3: How LLMs are trained

Building an LLM happens in stages:

1. **Pre-training.** The model reads a huge body of text and learns to predict the next token. This is where it picks up grammar, facts, reasoning patterns, and also the errors and biases in that text. It is by far the most expensive step.
2. **Fine-tuning.** It's trained further on carefully chosen examples of good answers, so it follows instructions and behaves like a helpful assistant instead of just continuing text.
3. **Learning from human feedback.** People compare pairs of answers and pick the better one. The model is adjusted towards answers people prefer: helpful, honest, and safe.

Some newer models also "**reason**": they work through a problem step by step before giving a final answer. That helps with maths, logic and code, but takes longer.

After training the model is fixed, with a **knowledge cut-off**. Some chatbots add a web search step so they can pull in recent information.

Chapter 4: Asking good questions (prompting)

What you type is called a **prompt**. Better prompts get better answers:

- **Be specific.** Not "write about saving", but "explain three ways a college student in India can save ₹1,000 a month, in simple English".
- **Give context.** Who is it for? What do you already know? What is it for?
- **Ask for a format.** A table, five bullet points, a 100-word summary, or Hindi.
- **Give an example** of what you want, if you have one.
- **Iterate.** If the first answer misses, say what to change. It's a conversation.
- **Ask it to show its sources or reasoning**, then check them yourself.

Never put passwords, OTPs, bank details or other people's private information in a prompt.

Chapter 5: Why LLMs make things up

Because an LLM generates *likely-sounding* text, it can produce **hallucinations**: fluent, confident statements that are false. Common ones:

- Invented facts, dates and numbers
- Fake quotes, fake news stories and made-up references or links
- Wrong legal, medical or financial details stated with total confidence
- Maths and counting mistakes

Hallucinations are more likely when the question is about something rare, very recent, very local, or after its cut-off.

How to protect yourself:

- **Check important facts** against an official or trusted source.
- **Click every link and reference** it gives. Fake ones are common.
- **Ask "how sure are you, and how could I verify this?"**
- **Don't use it as the final word** on health, law, money or news.

The rule: Use an LLM to draft, explain and brainstorm. Use reliable sources to decide.

Chapter 6: Choosing an LLM

There are many LLMs, and the right one depends on the job:

- **Closed models** (like GPT, Gemini, Claude) are used through the company's app or service. They're usually the most capable and easiest to use.
- **Open-weight models** (like Llama, Mistral, DeepSeek, Qwen, Sarvam) can be downloaded and run on your own computer or server, so your data stays with you. They need more technical skill.
- **Bigger isn't always better.** Smaller models are faster and cheaper, and are often good enough for simple tasks like sorting or summarising.
- **Language matters.** If you work in an Indian language, test models in that language. Some, like Sarvam's, are built specifically for Indian languages.

Benchmark rankings change every few weeks; try the models on *your* real tasks.

Chapter 7: LLMs and your data

When you type into an online chatbot, your words are sent to the company's servers:

- Many free services may use your conversations to **improve their models** unless you switch that off in the settings.
- Chats may be **stored** and, in some cases, reviewed by people for safety.
- Where the servers are matters. Some services store data outside India.

Good habits:

- Treat a chatbot like a **public notice board**: don't share anything you wouldn't want seen.
- Don't paste **confidential work documents** unless your employer has approved that tool.
- Check *Settings* → *Data controls* (or similar) and choose what's right for you.
- For sensitive work, consider an open-weight model that runs entirely on your own machine.

Key words

- **LLM:** a large language model, trained on huge amounts of text to predict the next token.
- **Token:** a chunk of text, roughly $\frac{3}{4}$ of an English word.
- **Context window:** how many tokens the model can consider at once.
- **Pre-training / fine-tuning:** the big general training step, then the focused one.
- **Prompt:** what you type to the model.
- **Reasoning model:** a model that works step by step before answering.
- **Hallucination:** confident, fluent output that is false.
- **Open-weight model:** a model you can download and run yourself.

Take the course for what you just read

Take the free 10-question assessment and pass with 8 or more. Every course you pass counts towards a Pathway Certificate issued by Shravya Digital, earned when you pass all five courses in the pathway: <https://digital.shravyaworld.org/courses/large-language-models>

Sources

Checked against these primary sources on 8 October 2026. Rules change: if something here looks out of date, trust the source.

- Google: Machine Learning Crash Course, Large language models (<https://developers.google.com/machine-learning/crash-course/llm>)
- IBM: What are large language models? (<https://www.ibm.com/think/topics/large-language-models>)