

What Is an LLM? How Language Models Generate Text – and Why They Hallucinate

Author: Zsolt Lengyel, iTStudy Hungary Training and Research Centre

Reviewer: Mária Hartyányi, iTStudy Hungary Training and Research Centre

Learning objective

Chatbots such as ChatGPT, Claude or Gemini answer in fluent, confident language – even when they are wrong. The aim of this material is to explain, with concrete examples, what a large language model (LLM) is, how it produces text-based answers, and why it can generate inaccurate, invented or misleading information. By the end of the learning content, teachers will be able to define the concept of “token”, to explain the principle of the next-token prediction, recognise the typical forms and warning signs of hallucination, and use practical strategies and tools for validating AI outputs..

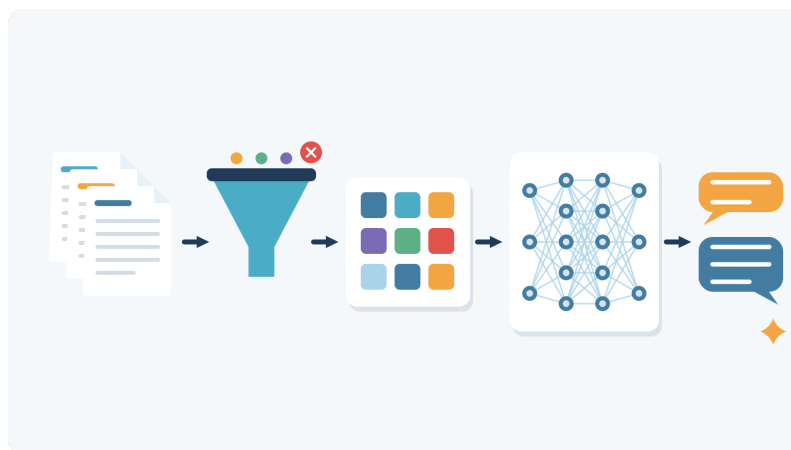
Introduction

On 30 November 2022 the first public version of ChatGPT was released, within five days a million people had signed up and since then, large language models have become everyday working tools of people. Their answers are well structured and confident. This is exactly why their errors are dangerous: a wrong answer looks just like a right one.

Teachers do not need the mathematics behind the Large Language Models, but a basic picture of how an LLMs work explains a great deal: why it is so good with languages, why it sometimes invents things, and why checking its output is a professional responsibility.

How a large language model is made

A large language model is an AI system trained on an enormous amount of text, whose main ability is to interpret and generate language. It is “large” because it contains hundreds of billions of parameters – adjustable numbers that store what the model has learnt. Building one takes four main steps:



From filtering web text through tokenisation and training to a chat assistant: the main steps of building an LLM

Collecting and filtering data

Developers download a large part of the public web and remove spam, duplicates, low-quality pages and personal data, leaving tens of terabytes of text. The model can only “know” what was in this filtered data – and only up to a certain date (the *knowledge cutoff*).

Tokenisation

The model does not see letters or words, but numbers. **Tokens** are word fragments, words or punctuation marks, each with an identification number. The vocabulary of a typical model contains about a hundred thousand tokens. A long Hungarian word is often split into several tokens – one reason why LLMs are surprisingly bad at counting a letter in a word.

Pre-training

The model receives a sequence of tokens and tries to predict the next one, then checks whether it was right and slightly adjusts its parameters. This is repeated trillions of times. The result is a very sophisticated “text completer”, but not yet an assistant.

Post-training

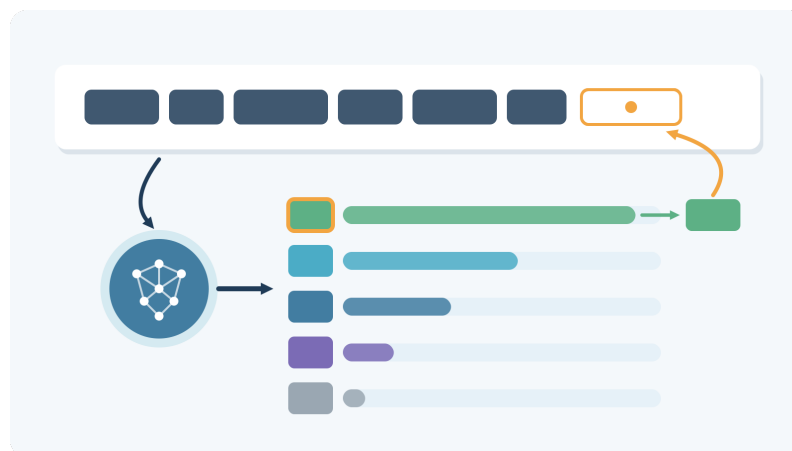
After it the model is then trained on carefully written questions–answer examples, so that it learns to answer helpfully, politely and in a useful format – and when to refuse. Finally, people or evaluator models rank alternative answers, so the model learns which answers are preferred.

Generating an answer: one token at a time

When we send a message, the model calculates for every possible next token how likely it is to follow the text so far. It picks one, adds it to the text and repeats the process – token by token. What we see on the screen as “typing” is the arrival of these tokens. Take the start of a workshop sentence:

“Before cutting the steel pipe with an angle grinder, always put on your ...”

Continuations such as *safety goggles, gloves* or *ear defenders* are highly probable, because similar sentences appeared many times in the training data; *slippers* are extremely unlikely. The model does not look up a rule – it continues the text in a statistically plausible way.



Next-token prediction: the model scores the possible continuations and adds one of them to the text

Three consequences are worth remembering:

Answers vary

The model does not always choose the single most probable token, so the same question can produce different answers.

The model needs tokens to “think”

Asked to give only the result, a model once answered that three machines, which need 12, 8 and 6 hours alone, would finish a job together in 3.43 hours. Allowed to work step by step, it reached the correct 2 hours 40 minutes. “Thinking” models generate such a line of reasoning before the answer.

The model does not remember like a human

Every reply is generated from the text it receives at that moment (see M1-T1-MLC5); current information comes only from tools such as web search, code execution or attached files.

Tokenization is the process of breaking up text into small units and then assigning a unique identification number to each unit. For example, in GPT-4, this “dictionary” – a list of all possible tokens – consists of 100,277 items. So, when we get to the model, our message will be a sequence of numbers – for example, [15339, 1917, 264, 3918]. The generation of the response works in the same way, but in reverse order: the model returns tokens one after the other, and then a simple program converts them back into readable text. We see it as continuous typing on the screen, but in reality, tokens arrive one after the other.

Let's try it!

Tokenization can be tested on an interactive site called Tiktokenizer (<https://tiktokenizer.vercel.app/>). Entering a text, we will see how it is broken up into tokens by the GPT 4o model, which word fragment will be an independent token and which will not.

Why AI hallucinates

The misleading phrase “AI lies” is often heard. Indeed, it happens that we get the wrong answer to a question, but this comes from the way the model is trained. It follows from the way the model was trained: in its training examples, questions are followed by confident answers, so the model learns to produce an answer-shaped text even when it has no reliable knowledge.



Humans hallucinate through their imagination! (Source: ChatGPT)

It is perhaps fortunate that the term **hallucination** has become established in the literature, even though the underlying process is fundamentally different from human hallucination: **AI does not possess the human capacity for imagination; it does not “see” or “imagine” things that are not there, but generates plausible information based on learned patterns—information that can sometimes be incorrect.**

In the context of Large Language Models, hallucination refers to AI-generated content that appears fluent, coherent, and plausible but is factually incorrect or unsupported by reliable evidence. It occurs because the model generates text by predicting statistically likely sequences of words rather than by verifying its statements against reality.

OpenAI researchers have also shown that common training and evaluation methods reward guessing more than admitting uncertainty (Kalai et al., 2025). Newer models say “I don’t know” more often, but the problem has not disappeared. A hallucinated answer is statistically consistent with real answers – same tone, same format – it is just not true.

Typical examples

Invented facts and people.

Asked about a person who is not well known, it happens that the chatbot gives a detailed biography, with occupation, works, years, which is completely randomly compiled.

Fabricated references and regulations.

Asked for the legal basis of a safety rule, a model may cite a decree with a number, date and paragraph in perfect format – which does not exist or regulates something else. In 2023 a New York court fined lawyers who had submitted a brief containing court decisions invented by ChatGPT (Mata v. Avianca).

Plausible but wrong technical values.

A tightening torque for a bolt, a cable cross-section, a cooking temperature or a tolerance may sound right because it falls within the usual range – but differs from the manufacturer’s or the standard’s value. Calculations can go wrong easily!

The risk of silent substitution.

The model may mistake rare technical terms for typos and replace them with similar, more common words. For example, it may automatically substitute an unfamiliar name of a plant, animal, or machine component with a familiar word that looks or sounds similar. The resulting answer may still appear fluent and convincing while heading in a completely different and incorrect direction.

Omissions and bias.

Asked how to cut a metal pipe, an AI may correctly suggest an angle grinder with a cutting disc – and say nothing about eye, hand and hearing protection. Models can also reproduce stereotypes from their training data.



A hallucinated answer can look official and complete – until someone checks the source behind it

Hallucination is more likely when the topic is rare, local or recent (national regulations, company procedures, last month's changes), when precise numbers, names or citations are required, and when the question contains a false assumption.

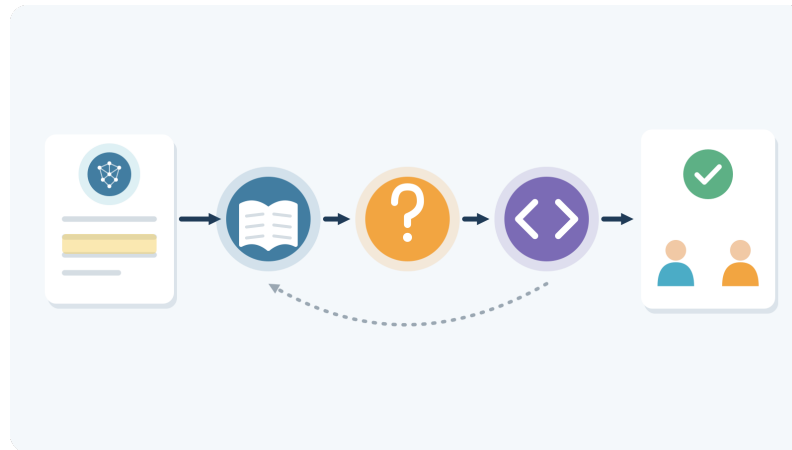
Put this question to ChatGPT:

Group the main fields and task types according to how reliable ChatGPT's answers tend to be. Create a table showing the estimated reliability range, the main risk of error, and the level of verification required. Treat percentages as measured results only when supported by reliable research; otherwise clearly label them as estimates.

The table provided in response offers important guidance for the informed use of AI. For example, it suggests that the probability of hallucination is relatively low for programming tasks (5–15%), but may reach as high as 60% when dealing with current information without access to web sources.

Detecting errors and validating outputs

Fluency and confidence are not signs of correctness. The right question is not “does it sound right?” but “how do I know it is right?” The following strategies help:



Check the source, ask follow-up questions and use tools – and only then take the content to the classroom

Check sources!

Ask for sources, then open them: does the document exist, and does it really say what the answer claims? Compare with authoritative references: the manufacturer's manual, the standard, the textbook, the curriculum.

Ask follow-up questions!

"Which parts of your answer are you unsure about?", "Where does this value come from?", "What would change if...?" Ask the same question again in a new chat or in another model: inconsistent answers are a warning sign.

Make it use tools!

For fresh facts, ask for a web search, for mathematical and technical calculations, please "use code". In this case, the AI will write a program to solve the problem and run it to provide the answer, which significantly reduces the possibility of errors..

Ground it in your documents!

Attach the relevant chapter or data sheet and instruct: "Answer only on the basis of the attached document; if the answer is not in it, say so."

Validate before classroom use!

Use your own professional knowledge – or a colleague's – to check every fact, number and safety instruction that learners will rely on.

Five checks before classroom use

1. Which statements are facts, numbers, names or references – and have I checked each of them?
2. Is anything safety-critical? Does it match the manufacturer's data and the current rules?
3. What is missing (protective equipment, preconditions, exceptions)?
4. Are the level, terminology and context right for my learners?
5. Would I sign this material with my own name?

A pedagogical and ethical issue

Hallucination is not simply a technical inconvenience. When a teacher hands out an AI-generated worksheet, its errors carry the teacher's authority, and learners – who are still building their professional knowledge – are the least able to spot them. In a workshop, a wrong value or a missing safety step can lead to damaged equipment or injury. Errors that learners memorise are hard to unlearn.

The responsibility therefore stays with the teacher: AI can prepare a draft, but the teacher validates it. It is equally important to teach learners to do the same – to ask where information comes from, to compare it with reliable sources and to treat AI as a starting point, not as the final authority. This is part of the AI and information literacy described in DigComp 3.0.

A bit of history

Since the invention of writing, humanity has continually sought to preserve, pass on, and make accumulated written knowledge accessible to future generations.

The emergence of writing, around 3200 BCE in Mesopotamia, made it possible for knowledge to outlive human memory. Cuneiform clay tablets were durable, but they could be read only by a small, trained class of scribes. Papyrus and parchment scrolls made written knowledge more portable and enabled the creation of the great libraries of antiquity, yet access remained largely restricted to a privileged few.

In the Middle Ages, the scroll was gradually replaced by the codex—the bound, page-based form of the book. Books copied by hand in monasteries were extremely costly and time-consuming to produce, so books and the knowledge they contained remained scarce. The printing press brought a fundamental change in the mid-15th century: texts could be reproduced in large numbers and at decreasing cost. The spread of books, followed by public libraries, mass education, and rising literacy, steadily expanded the number of people who could directly access accumulated knowledge.

At the end of the 20th century, computers and the internet brought another major leap. Digitization and global networks made enormous amounts of information available almost instantly. Many physical barriers disappeared: today, a single smartphone can provide access to resources that once required libraries, travel, or institutional privileges. But this created a new challenge. The problem is no longer primarily a lack of information, but navigating its abundance, understanding it, and distinguishing reliable information from unreliable content.

Large language models represent another step in the democratization of access to knowledge. The internet gave people access to information; LLMs go further by providing a natural-language interface for working with that information. Users no longer need to know exactly which book or database to consult, which technical term to search for, or even how to formulate the perfect search query. They can ask questions in everyday language, ask follow-

up questions, request simpler or more detailed explanations, examples, summaries, comparisons, or translations. The same subject can be explained to a beginner or a specialist, at different levels of detail and in different languages.

This is one of the most important democratizing effects of LLMs: **they can lower not only the barriers to finding information, but also the linguistic, technical, and search-related barriers to understanding it.** Earlier technologies progressively put books, and later information, within the reach of more people. Language models add a new dimension: they can adapt information to the user's question, language, context, and level of knowledge.



Preserving human knowledge: from clay tablets to language models (Source: ChaGPT)

This also marks an important difference from earlier forms of knowledge. A book preserves a fixed text; an LLM generates a new response to each question based on patterns learned during training. This makes it extraordinarily flexible, but not infallible. It can produce inaccurate or even entirely incorrect answers, and the origin of a statement is not always traceable to a specific source. **As access to knowledge becomes easier, the ability to evaluate and verify the answers we receive becomes increasingly important.**

From the clay tablet through the printed book and the internet to the LLM, we can therefore see a long process in which **knowledge has become accessible to more and more people, with fewer and fewer barriers. The distinctive contribution of LLMs is that the next stage of democratization is no longer only about access to information—it is increasingly about access to understanding.**

Takeaways for teachers

- An LLM generates text token by token, choosing plausible continuations based on patterns in its training data.
- It does not look up facts: fluent, confident answers can be wrong.
- Be especially careful with numbers, references, regulations, rare terms, local or recent topics and safety instructions.

- Check sources, ask follow-up questions, use search and code, and provide your own sources.
- Validate before classroom use – and teach learners to verify AI outputs, too.

References

- Kalai, A. T., Nachum, O., Vempala, S. S. & Zhang, E. (2025): Why Language Models Hallucinate. arXiv:2509.04664.
- Huang, L. et al. (2024): A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Transactions on Information Systems. DOI: 10.1145/3703155.
- Mata v. Avianca, Inc., 678 F. Supp. 3d 443 (S.D.N.Y. 2023).
- Cosgrove, J. & Cachia, R. (2025): DigComp 3.0 – European Digital Competence Framework. 5th ed., Publications Office of the European Union.
- Lengyel, Zs. (2026): Introduction to the concepts of artificial intelligence; Concrete applications for content generation. VITA training presentations.

Links

- Tiktokenizer – see how text is split into tokens: tiktokenizer.vercel.app
- Interactive 3D visualisation of an LLM: bbycroft.net/llm
- A. Karpathy: Deep Dive into LLMs like ChatGPT (video): youtube.com/watch?v=7xTGNNLPyMI

Copyright © 2026 VITA Consortium

This work may be reused and adapted under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).



Funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency.