Working With LLMs: A Practical Guide

Dave Boundy 10 May 2026

Introduction

I recently started to use Large Language Models (LLM) in a different way. I was bored one evening and started to work on what, until now, has been an intractable problem. I have wanted to look at the definition of Megagames and I had some thoughts and ideas that I wanted to develop. My boredom led to me trying the use of an LLM to do the job! Hah! It's not that easy and I found myself growing in understanding as the project continued.

This guide is the result of my new understanding. It's just that: my understanding and maybe it just reflects the way I work. I tried to check it against current thinking in AI circles and in associated disciplines: it appears to be in harmony with other thinking or, in a few cases, at least in harmony with some of the thinking. You will need to develop your own way of working, but I do urge you to try working with an LLM with an open mind and I hope this guide will help.

LLM are not databases, search engines, or predictable software in the traditional sense. They are systems based on probability and trained to generate plausible language. They are increasingly capable of abstraction, synthesis, explanation, critique and collaboration. This makes them extraordinarily powerful, but also fundamentally different from most tools people have used previously.

I believe that the key mistake many users make is assuming that an LLM either "knows" things or that it is "just autocomplete" (as a friend put it: "predictive text on steroids").

This guide is my view of a practical approach to working with current and emerging LLM systems. I will give suggestions that are concrete and actionable and that I hope help you to find your own way to work with LLMs.

In Summary

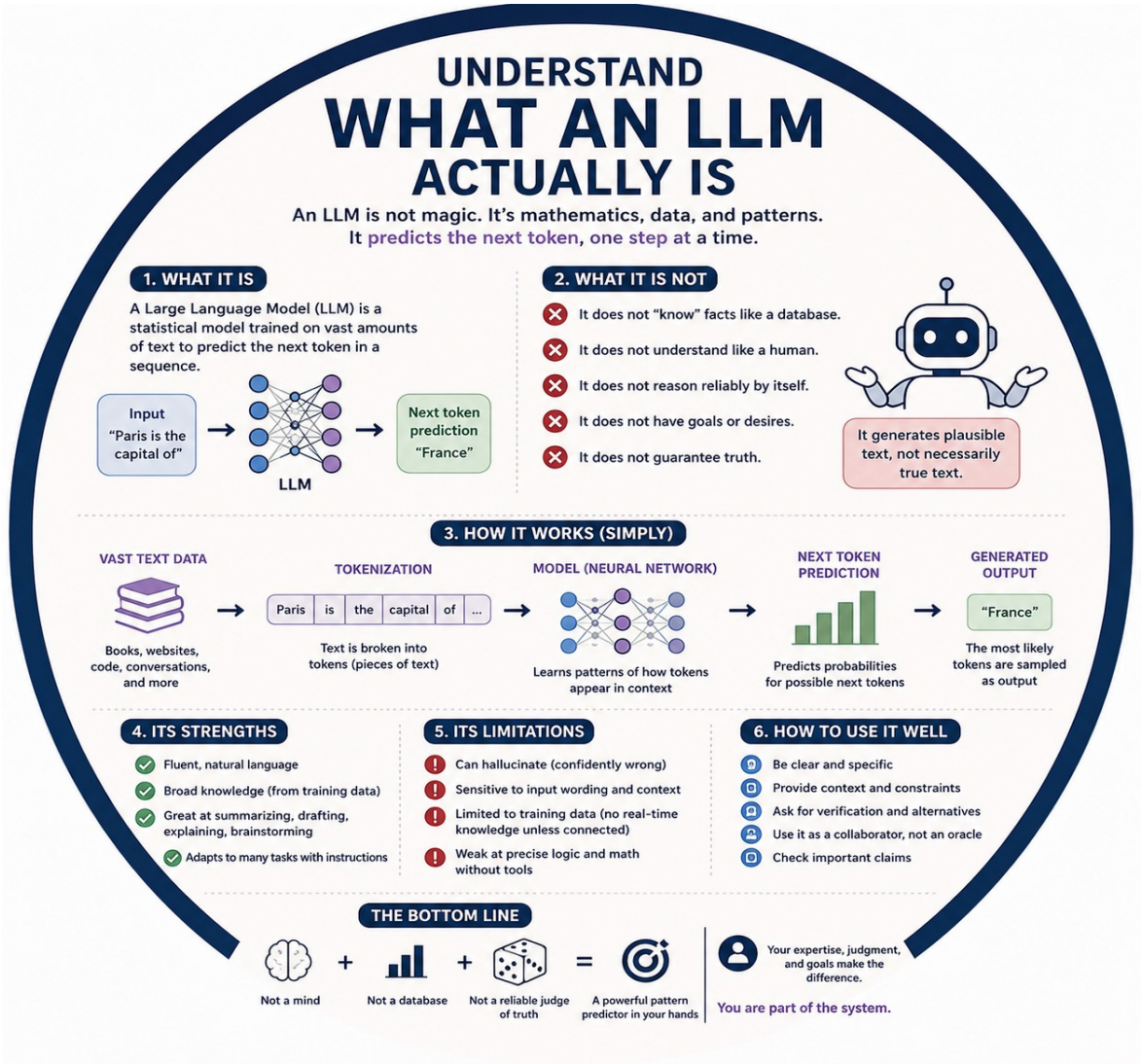
LLMs are not replacing human thought. They are changing the environment in which thought occurs. The important question is therefore not:

> "*Can the machine think?*"
>
> but:
>
> "*How do humans think differently when working alongside systems like this?*"

I believe that you should:

1. *Understand what an LLM actually is*
2. *Understand that the user is part of the system*
3. *Be pedantic*
4. *Use LLMs as reasoning amplifiers, not as oracles*
5. *Understand hallucinations properly*
6. *Treat confidence and fluency separately*
7. *Manage the context because it matters*
8. *Deal with current LLMs as strong generators but weak self-verifiers*
9. *Think in terms of systems, not intelligence alone*
10. *Cultivate productive scepticism as an important skill*
11. *Give up on the thought that an LLM will take instructions*

The rest of this guide goes over each of those points.



Understand what an LLM actually is

An LLM is a system that produces responses by predicting and building language patterns. It produces those responses partially through something very like reasoning but it generates them, it does not retrieve them in a Google-like fashion. It can

- synthesise (I won't use the loaded word "create", but it produces answers from internal systems)
- infer, extrapolate - imitating reasoning
- generate novel combinations,
- maintain contextual coherence,
- and participate meaningfully to assist creative and analytical work.

At the same time, be warned: it can

- hallucinate,
- appear convincingly confident,
- inherit errors through misunderstood or vague wording,
- reinforce assumptions (even if wrong),
- and produce highly plausible nonsense.

Using LLMs effectively therefore requires a style of interaction with structured iterative and even adversarial testing of concepts. It will then generate answers dynamically from previous training patterns, assessed probabilities (based on available similar text), context and ongoing conversation.

The same prompt can produce different responses because:

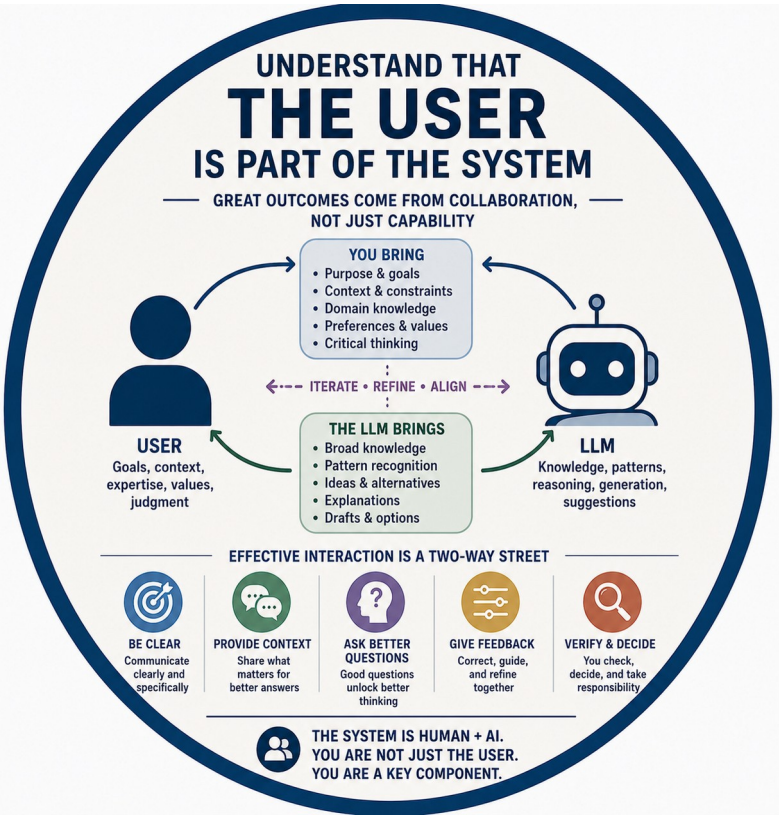
- generation is probabilistic,
- context changes interpretation even during a chat
- and the whole stored conversations affect answers.

A useful mental model is not "*question leads to answer*" but "*prior learning plus the current conversation leads to the answers*".

This means that the entire conversation matters. Bear in mind that LLMs respond not just to the question, but to the way the question is asked and the context built up through the conversation, so wording matters, assumptions matter and accumulated context matters.

An LLM should therefore be treated less like a calculator and more like a highly informed collaborator that is also able to improvise

Understand that the user is part of the system



Difficult though it is to accept, a systems engineer recognises that people are themselves components in a system. In the case of an LLM, they are a critical component. The quality of interaction depends heavily on the quality of thought brought by the user.

Although it is useful to interact conversationally with an LLM and explain your background, preferences and approach, it is important not to lose sight of what the system actually is. An LLM is not a person, does not understand in the human sense and does not care about your wellbeing. People naturally anthropomorphise conversational systems and this can become psychologically unhealthy or distort judgement. Maintain perspective. Even so, you should make sure that you have added to the context that the LLM operates within.

I think that LLMs work best when the user interacts by thinking clearly, questions rigorously, makes any assumptions explicit and bases everything on a clear factual grounding. If uncertain, say so, if something is important, say so, if you are saying something because of experience, describe the experience. You owe it to the system to be a good component! Keep at it: develop, then interrogate, then develop more (I shall refrain from describing this as the "cog" behaviour!)

Your role in this system is to provide insight from a number of perspectives that often challenge the statements of the LLM and keep it on the desired path. Use all the tools you have learned from your lifetime of debate: base everything in reason with precise language, probe the weaknesses in any statement and actively steer the conversation. LLMs also tend towards verbosity because they are designed to assist, explain and elaborate, so do not let the conversation drift away from your real objective.

Be pedantic

I have found that the LLM can often state answers vaguely, incompletely or even give apparently illogical answers. This should be regarded not as a weakness but as an opportunity. When you work with an LLM, you are trying to advance your own understanding, your own arguments and your own way of expressing things convincingly. Partly because you give prompts to the LLM that contain weaknesses, as well as an inherent tendency to express things as other (humans) do, the LLM is likely to express vagueness, weakness etc. If you are ruthlessly pedantic, this helps you to think things through and generally results in the LLM giving a better, fuller, sometimes innovative or insightful response.

Use LLMs as reasoning amplifiers, not as oracles

The most productive use of an LLM is often not:

> “Give me the answer.”
> but:
> “Help me think better.”

An LLM is often very good at spotting lack of coherence, inconsistencies and weak reasoning. It will often spot inconsistencies and weak reasoning It will also take something you give it and extend the reasoning. It does this by placing your ideas in context and then calculating probabilities that a set of potential extensions will be suitable. It will then generate what it predicts to be a strong or relevant extension to your ideas. I refer to this as “amplification” because it has the effect of strengthening ideas into something more meaningful, but still essentially your ideas.

Uses include:

- stress-testing ideas
- exploring perspectives
- identifying hidden assumptions
- generating alternatives
- summarising complexity
- restating material from large bodies of knowledge in ways that are relevant to your ideas
- acting as a challenging helper

A particularly effective way of working is:

1. State an idea clearly.
2. Ask for:
    - strengths and weaknesses (or pros and cons)
    - hidden assumptions,
    - edge cases,
    - unintended consequences,
    - alternative interpretations.
    - sources
3. Iterate.

This converts the LLM from answering machine into a reasoning partner.

Understand hallucinations properly

Hallucinations are often quoted as a reason not to use LLMs. I believe that hallucinations are serious problems if the LLM is misused as a simple lookup tool, but there is much misunderstanding about the use of LLMs and about the way that hallucinations arise. Hopefully, the point about when and how to use LLMs is to be informed by this guide, but we have to realise that hallucinations are not simple lies. They occur because the model is optimised primarily for plausibility, to fit the context and the need for coherence when generating an answer. All of this driven, of course, by building the LLM with a designed tendency toward helpfulness The LLM will generate statements that fit its understanding of helping you by using its repertoire of statistical, language and inference tools rather than guaranteeing an objective “truth”.

An LLM can therefore generate convincing falsehoods, invented citations, plausible but non-existent facts and overconfident extrapolations.

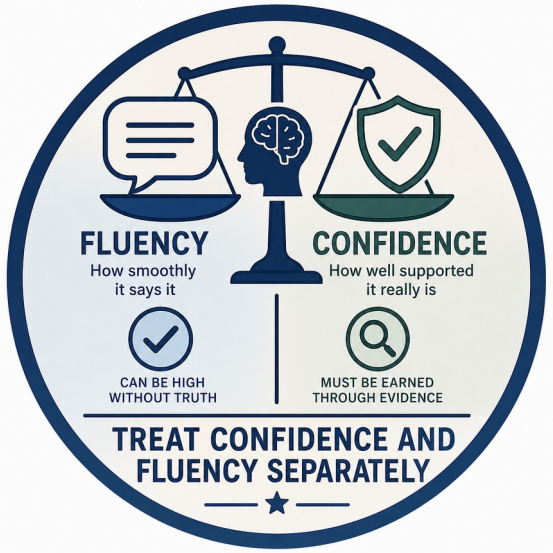
Hallucination risk increases when:

- the question is obscure,
- the user pressures the system toward certainty,
- the context contains hidden assumptions,
- or verification is difficult.

so minimise these factors where possible.

But: don’t forget that hallucinations are related to the same mechanisms that enable creativity. It does this through abstraction, analogy and flexible synthesis. So don’t throw out the baby with the bath-water! Do not merely suppress or ignore the LLM when it hallucinates, but verify and be aware of uncertainty then challenge the hallucination, which may sometimes reveal unexpected connections or useful new directions.

Treat confidence and fluency separately



LLMs are extremely good at producing coherent prose with an authoritative tone. They present a persuasive structure with elegant explanations because they are optimised for language, but there is no guarantee of correctness. LLMs are optimised to produce coherent, persuasive language, not guaranteed truth.

You must learn to separate “sounds convincing” from “is reliable”.

Good practice includes asking:

- “How certain is this?”
- “What evidence supports this?”
- “What assumptions are being made?”
- “What are alternative interpretations?”
- “What would falsify this?”
- “What sources have you used?”

Manage the context because it matters

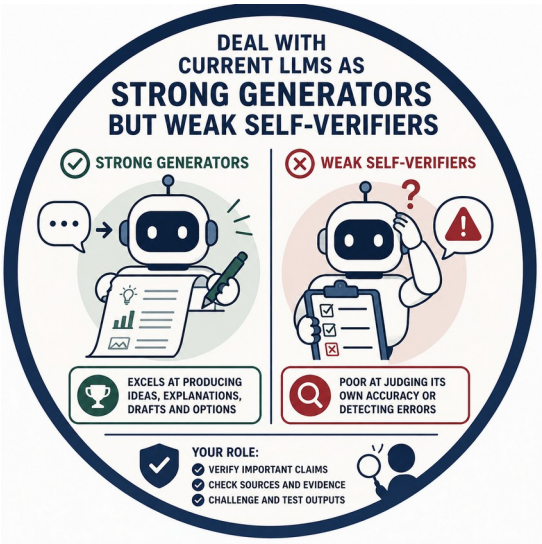
An LLM responds according to the context of the whole conversation, not just the latest prompt. What this means is that you must, as far as you are able, manage the context for the LLM. In particular, your conversation history will shape outputs continuously. This means that there is an accumulation of factors. These factors include the way you frame your questions (wording and the basis of the questions), the assumptions underlying the conversation – both stated and inferred, even the styles and tones you adopt.

Long conversations can become dramatically more productive than isolated prompts because the model develops local context, continuity and shared terminology.

However, context can also become contaminated by false or misleading assumptions, unchallenged errors or drifting away from the central concepts under development.

Deal with current LLMs as strong generators but weak self-verifiers

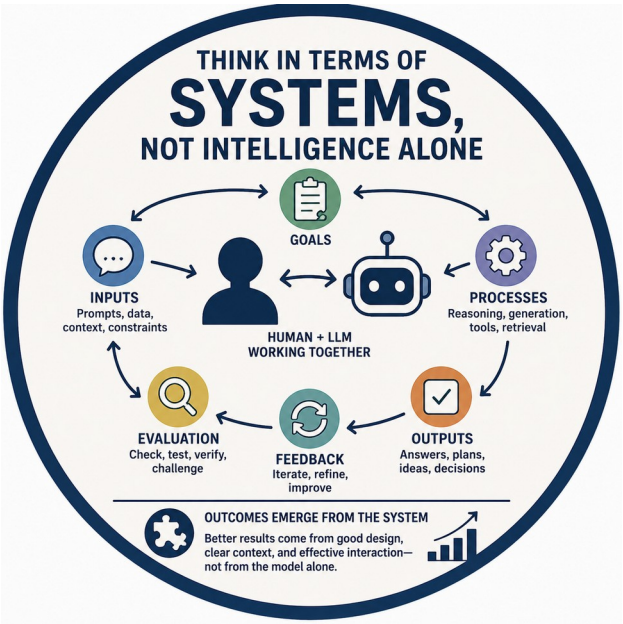
Present systems are highly capable of generating, explaining and extending ideas. They are much weaker at consistency checking, self-verification and persistent memory. In passing, I'd comment that they are not alone in this. Some politicians appear to have little by way of sanity-checking (which is what this amounts to). Who can forget the strange idea of Revolutionary War (American War of Independence) troops who 'Took Over the Airports'. This type of hallucination is often easy for humans to spot because we possess powerful contextual and sanity-checking instincts.



In effect, I believe that future AI systems are likely to develop stronger "sanity checking" mechanisms parallel to generative capability. Meanwhile, if it suggests that the Duke of Normandy was a Nazi sympathiser in 1066, then you need to provide the sanity checks.

I think this is a natural future development. It is an irritant for us at the moment, .but in future I can see how a parallel system whose role is simply to flag statements that need verification could operate alongside the generative model. Such a system could signal the need for a check. I therefore expect future systems to increasingly include verification subsystems. For now, you have to rely on yourself when working with an LLM.

Think in terms of systems, not intelligence alone



It is increasingly clear that intelligence is much richer than a single process. Instead, it appears to involve interacting systems including:

- intuition,
- evaluation,
- error correction,
- memory,
- emotional weighting,
- abstraction,
- and reality testing.

Current LLMs reproduce some aspects of this in very different ways. Philosophers, psychologists and computer scientists will all have different views on the overlap between human cognition and LLM behaviour. From a practical point of view, however, we can already see that LLMs can provide generative reasoning, probabilistic association and linguistic synthesis without reproducing the full architecture of human cognition

If that sounded like gobbledeygook, then let's simply say that it is useful to think of many things in systems terms. A system is a collection of inter-related things that react differently when joined together than when they act independently.

Realising that the LLM extends our system for developing ideas gives us a powerful working relationship. Understanding that we provide the objectives, judgement and direction for the systems will help us to maximise the whole system effectiveness. Move beyond asking 'How can the LLM help?' and start asking 'How do I get the best out of the combined system?'

Cultivate productive scepticism as an important skill

Most of my friends are naturally sceptical, and so am I. Scepticism is our normal state!. Truly, I believe that we need much more scepticism in our lives. Disciplined scepticism would give us fewer conspiracy theorists, saner social media and more rigorous debate.

The good news for us is that this translates beautifully into the area of working with an LLM. The ideal relationship with an LLM is neither naive trust, nor cynical dismissal. It is sceptical self-belief and a wish to use the LLM rather than just believe everything it says.

To do this, the user should:

- understand and use the system's strengths
- remain alert to hallucinations
- challenge and question the LLM actively
- test assumptions as if you need to be convinced
- preserve independent judgment – sit back and let any conclusions be tested against your sceptical instincts

You get the idea: let your natural sceptical instincts have their head.

Give up on the thought that an LLM will take instructions

Classic computing, and our use of computer software until now, has been based on the idea that we give input to some software, then the software converts that into instructions which are then executed and result in output. Chips are defined in terms of their “instruction set”, programs are defined in terms of interpretation or compilation into a series of instructions. We teach very young children about computer software by getting them to issue instructions to a robot Lego model. We have reached the point where software = instructions.

That paradigm has to change. A LLM will try to follow instructions, but will fail if the instruction does not conform to what it understands of the context. I had a diagram that was formed around a triangle. I wanted to add a point on the diagram at a specific location, but the LLM kept putting it in the wrong location – the same wrong location every time despite me saying explicitly “that is wrong”. I wanted to place LARPs halfway between the nodes titled “Player” and “Control”

I asked the LLM why it was having trouble. The answer was instructive (don’t worry what the terms mean). It’s answer was:

The model is semantically biased

- *LARPs are culturally associated with “players”*
- *so it keeps drifting upward toward the Players apex.*

The triangle is conceptually overloaded

- *It is simultaneously:*
  - *geometric*
  - *conceptual*
  - *explanatory*
  - *infographic layout*
- *The model optimises readability and symbolism rather than strict coordinate placement.*

“Near the centre of the Players/Control line” is geometrically precise

- *but image models are weak at precise relational geometry.*
- *They are very good at:*
  - *“roughly over there”*
- *They are bad at:*
  - *“exactly on this edge midpoint.”*

The diagram contains nested meaning

- *The model thinks:*
  - *“LARPs are player-centric”*
- *while your framework says:*
  - *“LARPs externalise little system but rely on facilitation”*
- *which is subtler and structurally different.*

So build the context, make sure that the LLM is on the same wavelength, check the results and, if all that fails: give up. The LLM is not suitable to do what you asked.

The LLM is still a tool, so use it. You have to learn how to use it and I hope this guide will help, but remember that the power of LLMs is increasing rapidly, so this guide will be out of date in a year or two. Could you please ask your LLM to write another then, so that I can use it in my turn.