Prompt Engineering and Effective LLM Interaction

A working guide for people new to Large Language Models (LLMs)

Sixteen named techniques, one worked example, and how to tell why an answer went wrong.

What we will cover

- What an LLM is, and how it differs from a search engine
- The six slots every good prompt fills
- The sixteen patterns, and the four to learn first
- One task done badly and then well — plus how to diagnose a bad answer

A prompt is not a search query

A question (Search Engine). “What should I include in a personal statement for graduate school?”
Returns a generic list — the only correct answer to a question with no specifics in it.

A prompt (LLM). “You are an admissions reader for a public-health master's. Ask me questions, one at a time, until you can draft my statement. Then draft 600 words and list the claims I must defend in an interview.”

A role, an interview, a length, and a checking step.

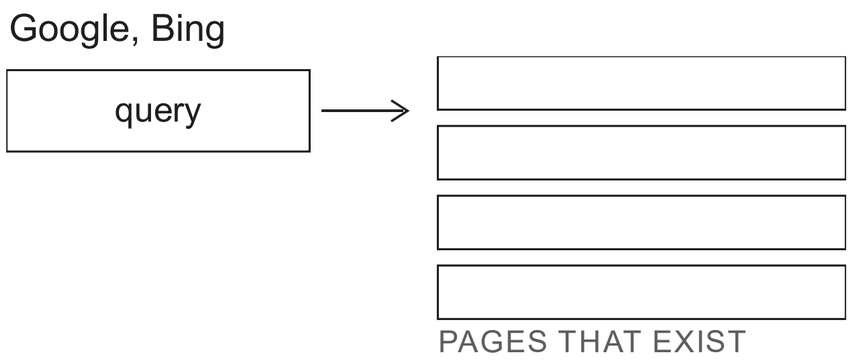
One mechanism: predict what comes next

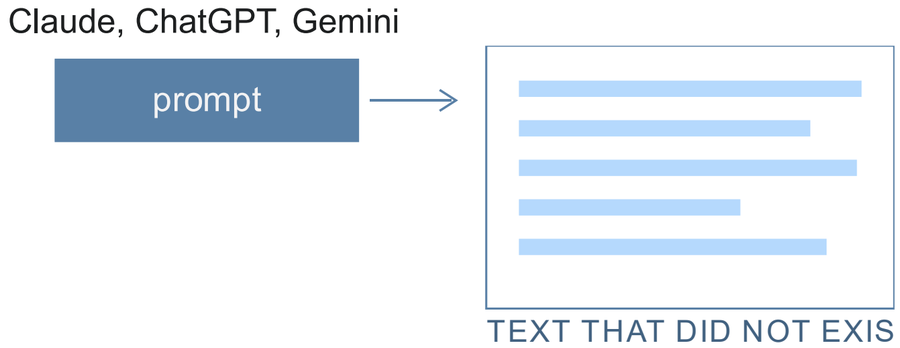
An LLM is trained on an enormous quantity of text to predict what text plausibly follows.

- **Why it is fluent.** Fluent continuation is exactly what it was trained to produce. Training optimizes one thing: select the next likely token. Training rewards text that *sounds* right — right register, right structure, no audible hesitation — and never asks whether the claim is true. So, it sounds equally confident whether it knows the answer or is guessing; there is no wobble in its voice to warn you.
- **Why it can be wrong.** Plausible and true are not the same thing, and there is no store of facts to check against.
- **Why prompts matter.** Your prompt is the start of the sequence being continued. LLM receives prompt as text, and then writes whatever text plausibly follows it.

Two Tools: One finds; One generates.

Not the same tool





A search engine finds text that already exists. You get ranked documents you can open and judge. It fails visibly — no results, or plainly irrelevant ones. Best for facts, sources and current information.

An LLM generates text that has never existed. There is no page behind the answer. It fails invisibly — fluent and wrong. Best for transforming, explaining, drafting and questioning material you supply.

SEARCH TO FIND · USE LLM TO WORK ON WHAT YOU FOUND

Six Terms worth knowing

- **Context window.** All the text it can attend to at once. Finite — and everything in it competes for attention.
- **Training cut-off.** Where its knowledge stops. Without search, anything newer is unknown — and it may answer anyway.
- **Hallucination.** Fluent invented content: a fabricated citation, a plausible wrong number. A by-product of generating, not a malfunction.
- **Tokens.** The chucks text is split into – roughly three quarters of a word each. Limits and pricing are counted in tokens, not words.
- **Same prompt, different answers.** Generation is deliberately random. “It worked once” is weak evidence that a prompt is reliable.
- **System Prompt.** Standing instructions set before your conversation begins – by you in settings or project field.

Basic prompt formula

GOAL — What outcome do I want?

CONTEXT — What does the LLM need to know?

CONSTRAINTS — What must or must not happen?

OUTPUT — What should the result look like?

Longer is not automatically better.

THE ANATOMY OF A PROMPT

Any slot you leave empty is something the model will silently invent.

1. **Role and task.** Who it acts as, and what it is doing.
2. **Context.** The actual material, the audience, the constraints, what is already decided.
3. **Definition of done.** What a good result achieves — not only what it looks like.
4. **Format.** In numbers. “Concise” is not a constraint; “under 400 words” is.
5. **Boundaries.** What to leave out, what not to change. Models add by default.
6. **Escape hatch.** “If anything is unclear or missing, ask before you start.”

Give an example, not an adjective. One forty-word sample beats three paragraphs describing it.

Sixteen patterns, five categories

INPUT SEMANTICS	OUTPUT CUSTOMIZATION	ERROR IDENTIFICATION	PROMPT IMPROVEMENT	INTERACTION & CONTEXT
Meta Language Creation	★ Persona	★ Fact Check List	Question Refinement	★ Flipped Interaction
	★ Template	Reflection	Cognitive Verifier	Game Play
	Recipe		Alternative Approaches	Infinite Generation
	Output Automater		Refusal Breaker	Context Manager
	Visualization Generator			

★ START WITH THESE FOUR

- From White et al., “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT”, arXiv:2302.11382.

The four to learn first

FLIPPED INTERACTION

It interviews you until it has what the task needs. Moves the ambiguity to the front.

> *“Ask me questions one at a time until you have enough to write this. Don't start until you do.”*

PERSONA

A role brings its concerns with it. Strongest as a reviewer, not an author.

> *“Read my report as the teaching assistant grading it against this rubric. Mark only what loses marks.”*

TEMPLATE

You give the skeleton; it fills the holes. Kills preamble, makes items comparable.

> *“Answer only in this form: AUTHOR (YEAR) — CLAIM — EVIDENCE TYPE — RELEVANCE.”*

FACT CHECK LIST

The claims the answer rests on, so you know what to verify.

> *“List every factual claim this depends on and mark the three you are least confident about.”*

Patterns combine

PERSONA (sets the standard the work is judged by)

+ FLIPPED INTERACTION (removes the ambiguity before anything is written)

+ TEMPLATE (returns something you can use directly)

+ FACT CHECK LIST (tells you what to verify before you rely on it)

One pattern fixes one problem. Several at once change what the whole interaction is.

— "As a staff engineer, interview me until you can propose the design; deliver it in the template below; end with the assumptions I need to confirm."

Five turns, wrong output

1. “Help me make a presentation about this article.” — a generic outline that fits any article.
2. “Make it shorter.” — the same outline, fewer words. The outline was never the problem.
3. “Add detail about the findings.” — confident detail, some of it not in the article, which was never supplied.
4. “My audience has already read it.” — a partial rewrite that still summarizes.
5. “Make it more interesting.” — livelier phrasing on the same wrong plan. Forty minutes gone.

• The task: fifteen minutes presenting an article to twelve classmates and a professor who have all read it.

Four separate failures — no material, no audience, no definition of done, wrong frame — **all treated with rewording.**

Four turns, usable output

- **Persona, context, escape hatch, Flipped Interaction.** The role, the real constraints, the full text, and “ask me questions before producing anything.” Three questions come back, all of which change the plan.
- **Alternative Approaches.** Three framings with trade-offs, nothing written yet. A structured objection beats a summary.
- **Template and Output Automater.** “Six slides: TITLE — ONE SENTENCE I SAY — EVIDENCE WITH PAGE — MINUTES.” A timed plan that adds to fifteen minutes.
- **Persona and Fact Check List.** “Read this as the professor. Where would you interrupt? List what I should verify.” Two challenges to prepare for.

What changed was the order, not the wording. Twenty-five minutes instead of forty.

A bad answer has a cause

Stop rephrasing and retrying

SYMPTOM	LIKELY CAUSE	FIX
Generic — fits anything	No material, audience or goal	Paste the source; say who it is for
Answers a different question	Ambiguity you could not see	“Restate the task, then wait”
Right content, wrong shape	No format specified	Template, with numbers
Ignored a constraint	Stated once, buried in the thread	Re-pin: “still in scope: …”
Confident facts that are wrong	Generating, not retrieving	Supply the source; Fact Check List
Quality fell off in a long thread	The window is crowded with dead ends	Start fresh with the current state

Correct the instruction, not the output. “Make it shorter” fixes one answer; naming the rule that was broken fixes the conversation.

It will agree with you if you let it
Sycophancy

Ask “is this any good?” and you will usually be told yes — after that question, praise is the plausible continuation. Change the question and you change what is plausible.

CASE AGAINST

“Make the strongest argument that this fails. Do not balance it.”

HIDE THE AUTHOR

“Two students submitted these. Assess both.”

CRITERIA, NOT TASTE

“Score this against the rubric, line by line, with evidence.”

A CRITICAL ROLE

“You are the reviewer who recommends rejection. Write your report.”

DOES IT CAVE?

Push back on an answer you think is right. If it folds, it had no real basis for it.

WHAT WAS LEFT OUT

“What would a skeptic point out that you omitted to be polite?”

What does not work

- **Politeness as a quality lever.** Clarity is the lever.
- **Threats, bribes, urgency.** Anecdotal at best.
- **“You are the world's greatest expert.”** Superlatives add nothing.
- **Longer prompts.** Relevance matters; padding competes for attention.
- **“Be concise.”** Not a constraint. Numbers are.
- **“Let's think step by step.”** Mattered in 2022; can hurt reasoning models now.
- **Asking for sources.** Citations can be invented. Check they exist.
- **Its stated confidence.** Generated text, loosely tied to accuracy.
- **Correcting it to teach it.** Applies to this conversation only.
- **Rephrase and retry.** Diagnose first, then fix.

Learn why a technique works and you will know when it has stopped being necessary.

Is this a good task to hand over?

WELL SUITED

Work that is quick to verify — a draft you would edit anyway, a summary of a text in front of you, practice questions, a restructuring of your own material.

POORLY SUITED

Anything you cannot check — unverifiable facts, citations you will not read, work that goes out without review.

Cost here means **your time, not money** : if checking takes longer than doing the task would have, the delegation was a loss. So set work up so checking is fast — page references, claims separated from prose, output you can test.

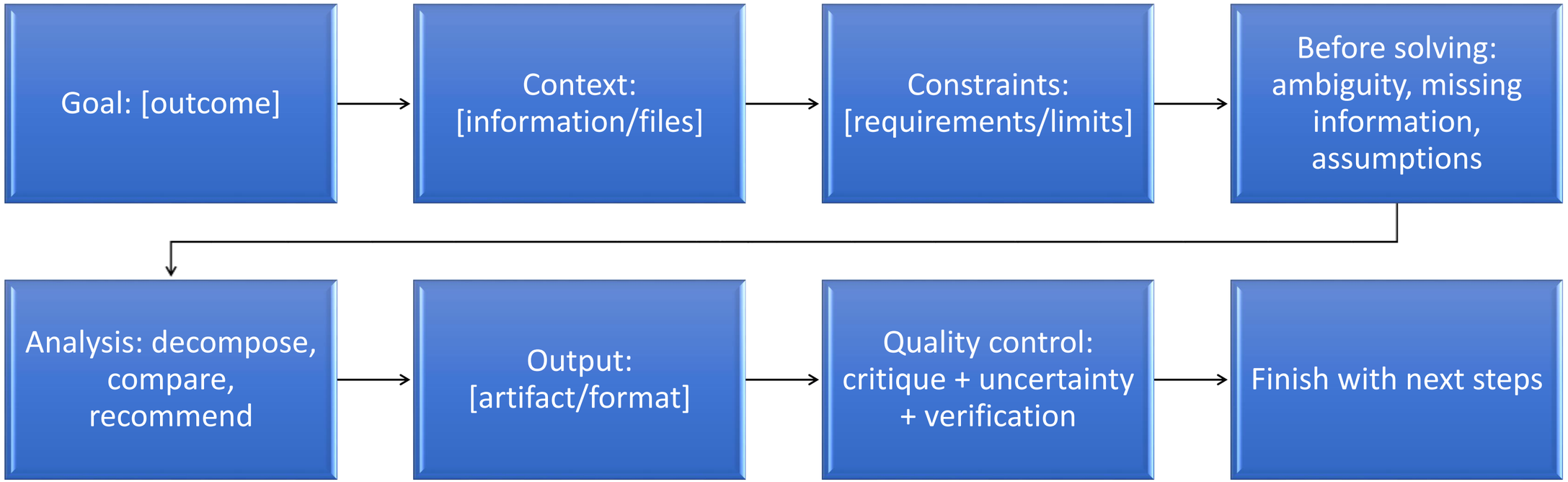
- **If you could not tell a good answer from a plausible one, you are not in a position to use the answer.**

Repeatable workflow

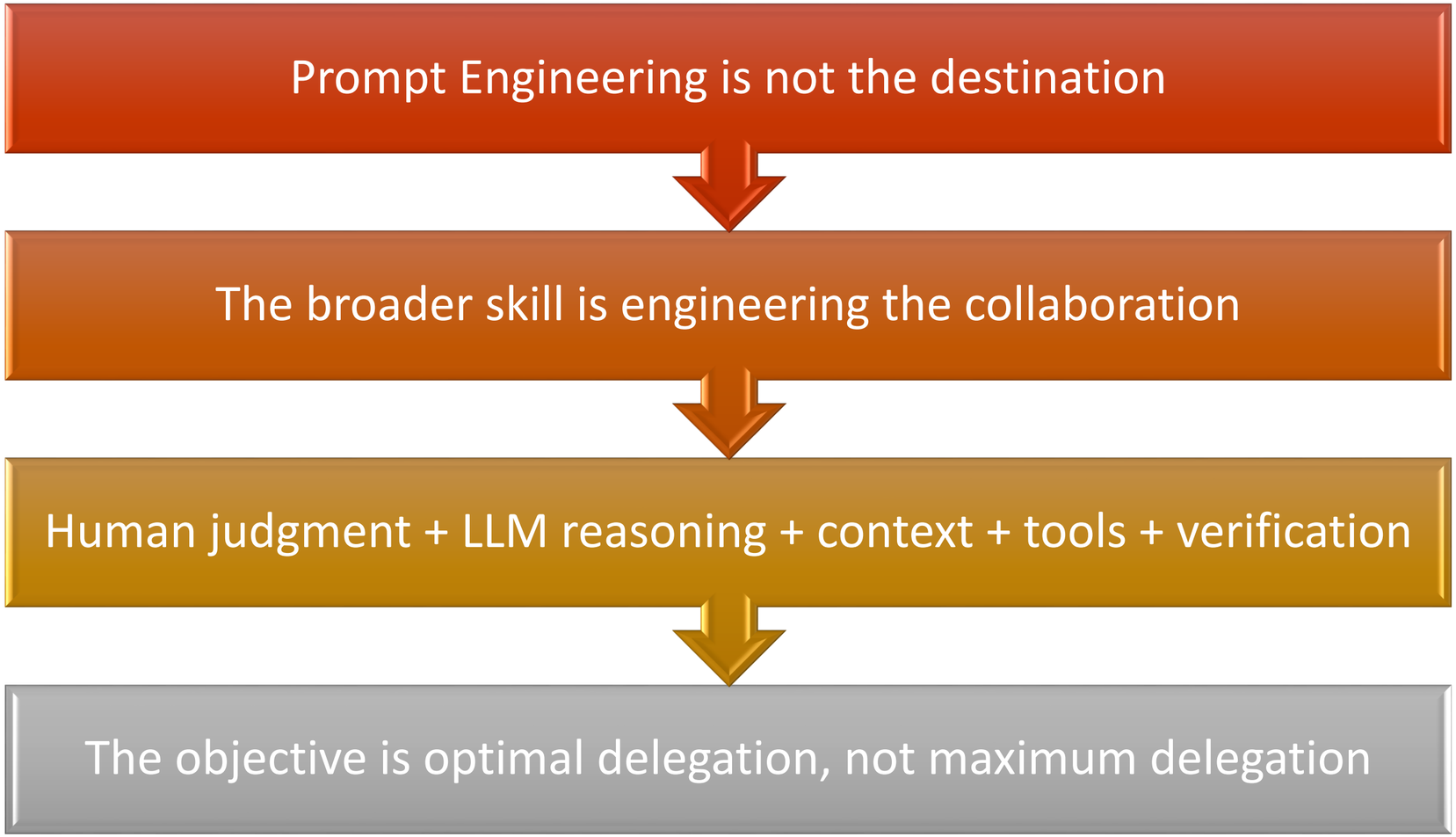
Ten questions to keep handy

- What am I trying to accomplish?
- What information is missing?
- What assumptions are you making?
- What am I not asking that I should be asking?
- What alternatives exist?
- What is the strongest argument against this?
- What could make this answer wrong?
- What should I verify independently?
- What would change your recommendation?
- What are the concrete next steps?

A practical master prompt



The bigger picture



If you remember four things

01 Supply the material. Most bad answers are missing context, not bad wording.

02 Settle the task before asking for output — let it interview you.

03 Say what “done” means: audience, length, format, what to leave out.

04 Build the check into the prompt. Never let fluency stand in for accuracy.

Write the prompt you should have started with

At the end of a task that went well, ask for it — and save the answer. That is how a prompt library actually gets built.

> “That worked. Write the prompt I should have given you at the start to get this in one shot — as a reusable template, with placeholders for the parts that change.”

Edit before you save it. The model’s reconstruction over-includes every correction you made and often drops the constraint that mattered.

Note the date and the model. A prompt tuned around one model’s blind spot is dead weight a year later.

Short version — 12 minutes

Full tutorial — 11 sections, sixteen patterns in full

After White et al., arXiv:2302.11382