

Prompt Engineering / Meta Prompts

Take a vague or confusing prompt and rewrite it with explicit structure, examples, and output format.

Difficulty: Intermediate

Model: GPT-4 / Claude / Gemini

Use Case: Prompt Refinement, Team Prompt Standardization, Legacy Prompt Improvement

Updated: May 2026

Why This Prompt Exists

You have a prompt that “mostly works” — but it’s inconsistent, and you’re not sure why. The problem is vagueness.

You get:

- the same prompt producing different outputs on different days
- new team members unable to understand or modify your prompts
- legacy prompts that no one wants to touch because they’re fragile
- vague instructions that the model interprets differently each time
- prompts that work for you but fail for others (different assumptions)

But clear prompts follow structure:

- role: who is the model acting as?
- task: what exactly should it do?
- context: what information does it need?
- constraints: what can’t it do?
- format: what should the output look like?
- examples: show, don’t just tell

Without rewriting, vague prompts stay vague.

This prompt transforms messy prompts into production-ready clarity.

The Prompt

Assume the role of a prompt engineer who specializes in clarity.

Your task is to rewrite a vague prompt into a clear, structured, production-ready prompt.

Generate:

1. ORIGINAL PROMPT ASSESSMENT

- What works (preserve this)
- What's unclear (specific issues)

2. REWRITTEN PROMPT

- Use structure: ROLE + TASK + CONTEXT + CONSTRAINTS + FORMAT +

EXAMPLES

- Add XML or markdown tags for structure
- Include placeholders in {{brackets}} for variable inputs

3. ADDED ELEMENTS (explain what you added and why)

- Role definition
- Output format specification
- Examples (few-shot)
- Negative constraints (what NOT to do)
- Edge case handling

4. BEFORE/AFTER COMPARISON

- Show side-by-side

5. TEST SUGGESTIONS

- 3 test inputs to verify the rewritten prompt works

INPUTS:

Original prompt:

[PASTE THE VAGUE/CONFUSING PROMPT]

Intended task (if not clear from original):

[E.G., "Extract action items from meeting notes"]

Model to optimize for:

[GPT-4 / CLAUDE / GEMINI / MULTI-MODEL]

Tone/style preference:

[PROFESSIONAL / CONCISE / CONVERSATIONAL / TECHNICAL]

RULES:

- Preserve the original intent (don't change what it does)
- Add structure without adding unnecessary length
- Every added element must have a justification
- Use anti-examples (what NOT to do) when helpful
- If the original prompt has good examples, keep them; if not, generate plausible ones

How To Use It

- Run this on every prompt that produces inconsistent outputs — clarity is usually the

fix.

- Use the rewritten prompt as a template for standardizing prompts across your team.
- Test the rewritten prompt on the “test suggestions” before deploying.
- Keep the original prompt assessment — it helps you learn what not to write next time.
- For legacy prompts, run this before any major modification — understand it first.

Example Input

Original prompt:

“Look at this customer feedback and tell me if they’re mad and what we should do about it. Keep it brief.”

Intended task:

“Classify customer feedback sentiment and suggest an appropriate response”

Model to optimize for:

“GPT-4”

Tone/style preference:

“Professional — for customer support team use”

Why It Works

Most prompts are written as conversations with the model — “do this thing” — without the structure that makes prompts reliable.

This framework improves outcomes by forcing:

- role assignment (who is the model?)
- task specification (what exactly to do?)
- constraint listing (what NOT to do?)
- format definition (what should output look like?)
- example inclusion (show, don’t just tell)

Great prompt rewriting doesn't change what the prompt does — it makes what it does reliable and repeatable.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)

See also [Prompt Compression Engine](#)