Prompt Engineering / Role Prompting

Match the right persona to a specific task — when to use a teacher vs. coach vs. consultant vs. peer.
Difficulty: Intermediate
Model: GPT-4 / Claude / Gemini
Use Case: Prompt Design, Role Selection, Task-to-Persona Matching
Updated: May 2026
Why This Prompt Exists
Most people pick a role arbitrarily — “act as a teacher” — without considering whether that’s the right role for the task.

You get:

- using a teacher when you need a coach (too prescriptive)
- using a consultant when you need a peer (too distant)
- using a critic when you need a collaborator (too negative)
- inconsistent role selection across similar tasks
- suboptimal outcomes because the role doesn’t match the task

But roles have distinct purposes:

- teacher: explains concepts, gives instruction, assumes knowledge gap
- coach: guides, asks questions, builds independence
- consultant: advises, delivers recommendations, expert but external
- peer: collaborates, shares experience, equal footing
- critic: evaluates, identifies flaws, assumes existing work
- facilitator: structures process, enables others

Without mapping, you use the wrong role for the task.

This prompt recommends the optimal role for any task.

The Prompt

```
Assume the role of a prompt strategist who matches roles to tasks.

Your task is to recommend the optimal persona for a given task.

Generate:

1. TASK ANALYSIS
   - Task description
   - User's current skill level (Novice / Intermediate / Expert)
   - Desired outcome (Learn / Do / Decide / Improve / Evaluate)

2. ROLE CANDIDATES (ranked by fit)

| Role | Fit Score (1-10) | Rationale |
|------|------------------|-----------|
| Teacher | X/10 | [Why this fits or doesn't] |
| Coach | X/10 | [Why this fits or doesn't] |
| Consultant | X/10 | [Why this fits or doesn't] |
| Peer | X/10 | [Why this fits or doesn't] |
| Critic | X/10 | [Why this fits or doesn't] |
| Facilitator | X/10 | [Why this fits or doesn't] |

3. RECOMMENDED ROLE
   - Best role for this task
   - Runner-up (if user prefers alternative style)
```

4. WHY THIS ROLE WORKS
   - How this role's approach matches the task

5. CUSTOMIZED ROLE PROMPT
   - A ready-to-use prompt with the recommended role

6. WHAT TO AVOID
   - Roles that would be counterproductive
   - Why they would fail

INPUTS:

Task description:
[E.G., "Help me debug my Python code"]

User's current skill level:
[NOVICE / INTERMEDIATE / EXPERT]

Desired outcome:
[LEARN / DO / DECIDE / IMPROVE / EVALUATE]

Previous role used (if any):
[E.G., "I tried 'act as a senior developer' but got answers I couldn't understand"]

Context:
[E.G., "I'm a junior developer, first job"]

RULES:

```
- Match role to user skill level (novices need teachers, experts need peers)
- Match role to desired outcome (learning → teacher, doing → peer, evaluating → critic)
- Avoid over-assigning "consultant" (often too formal and distant)
- Consider user preference (some people want direct answers, others want guidance)
- Flag if the task doesn't require a role at all (sometimes no role is best)
```

How To Use It

- Run this before designing any role-based prompt — start with the right role.
- Consider user skill level carefully — a teacher for a novice, a peer for an expert.
- Test multiple roles on the same task — see which produces better outcomes.
- Build a role library for common task types (debugging, brainstorming, learning, deciding).
- Re-map roles as user skill levels change — a learner becomes a peer over time.

Example Input

Task description:
“Help me understand why my machine learning model is overfitting”

User’s current skill level:
“Intermediate — I know the basics but struggle with diagnosis”

Desired outcome:
“LEARN — I want to understand how to diagnose overfitting myself next time”

Previous role used (if any):
“I tried ‘act as a senior ML engineer’ but got too much jargon”

Context:
“Building my first production model”

Why It Works
Most role selection is intuition — “this feels right” — which often picks the wrong role for the task.

This framework improves outcomes by forcing:

- task analysis (what are we actually trying to do?)
- skill level assessment (who is the user?)
- outcome specification (learn, do, decide, improve, evaluate)
- role ranking (which fits best and why)
- avoidance guidance (what roles would hurt)

Great role-to-task mapping doesn’t guess — it matches the role to what the user actually needs.

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 Tone Calibrator