

# Google's 69-Page Prompt Engineering Masterclass: What's Inside

*What's Prompt Engineering, Anyway?*

---

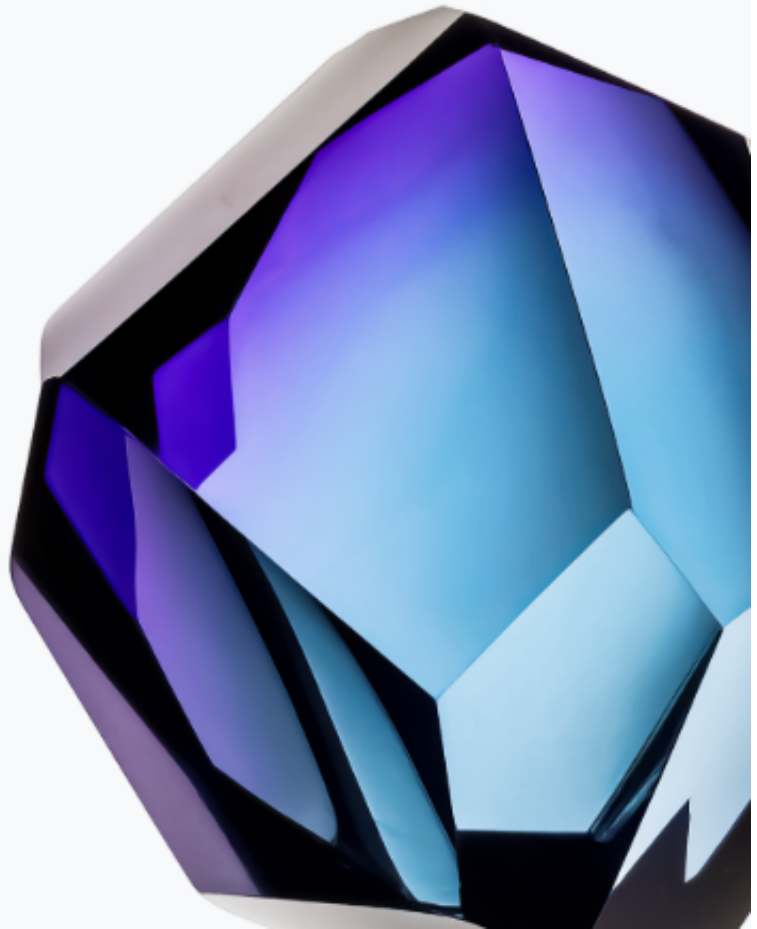
Algo Insights • 6 min read • 2025-05-03

<https://medium.com/coding-nexus/googles-69-page-prompt-engineering-masterclass-what-s-inside-099fa1466035>

---

# Prompt Engineering

Author: Lee Boonstra



I've been writing about tech for three years, and let me tell you, nothing's grabbed me quite like messing around with AI. These language models - fancy tech for "super smart chatbots" - can

crank out stories, code, or even math answers.

But here's the catch: you've got to ask them right.

That's where **prompt engineering** comes in.

It's not some nerdy science; it's just figuring out how to talk to AI without it going off the rails.

I leaned hard on Lee Boonstra's *Prompt Engineering* whitepaper for this, and I'm gonna break it down with simple examples so you can try it yourself.

Okay, imagine you're telling your dog to fetch a ball. If you just say "go get it," he might bring back a sock. But if you point at the ball and say "fetch that red one," you're golden.

Prompt engineering is like that - giving AI clear directions so it doesn't spit out nonsense.

The Google's whitepaper says these models guess the next word based on what they've learned. A good prompt nudges them to guess right. It's less about fancy words and more about being straight-up clear. Let's see how it works.

## Why Should You Care?

I tried asking an AI about "bears" once, hoping for wildlife facts. Nope - got a paragraph about the Chicago Bears. Total facepalm. A solid prompt stops those mix-ups, whether you want a poem or a grocery list.

You don't need to be a tech whiz for this. The whitepaper's all about anyone being able to jump in. It's like learning to text your friend so they actually get what you mean. Here's a few ways to do it.

## Cool Ways to Ask AI Stuff

The whitepaper's got a ton of tricks, but I'll stick to a few that don't make your brain hurt. Each one's like a different way to get your point across.

### 1. Straight-Up Asking (Zero-Shot)

This is the lazy way - just tell the AI what to do, no extras. Works for quick things.

**Try this:** Figuring out if a movie review's good or bad.

```
Hey AI, is this movie review POSITIVE, NEUTRAL, or NEGATIVE?  
Review: "Loved every minute of it!"  
Answer:
```

Thumbs-up

It's fast. But the whitepaper warns it can mess up if the review's tricky, like "awesome but boring."

### 2. Showing Examples (Few-Shot)

This is like saying, "Do it like this." Give the AI a couple examples, and it copies the vibe. I use it when I need something tidy, like data.

**Here's one:** Making a pizza order look organized.

Turn this pizza order into JSON.

Example 1: "Small pizza with pepperoni"

Output: {"size": "small", "toppings": ["pepperoni"]}

Example 2: "Large pizza with olives, mushrooms"

Output: {"size": "large", "toppings": ["olives", "mushrooms"]}

Now try: "Medium pizza with sausage"

Output:

Size: medium, Toppings: sausage

Examples keep the AI on track. The whitepaper says toss in a few - three's usually enough - to avoid screw-ups.

### 3. Setting Rules (System Prompt)

This is like telling the AI, "Put the answer in a box." It's great for making sure you get exactly what you need.

**Check this:** Getting a review all neat and tidy.

Classify this review as POSITIVE, NEUTRAL, or NEGATIVE, and return it as JSON.

Review: "It was fine, I guess."

Format: {"sentiment": "POSITIVE | NEUTRAL | NEGATIVE"}

Output:

```
{"sentiment": "NEUTRAL"}
```

It forces the AI to stay clean and simple. The whitepaper loves this for cutting down on random gibberish.

### 4. Playing Pretend (Role Prompt)

This one's fun - you tell the AI to act like someone, like a chef or a travel nerd. It changes how they talk.

**Try this:** Pretending the AI's your tour buddy.

You're a travel guide. Name 3 must-see spots in Rome.

Go:

- Eiffel Tower: Gotta see that giant sparkly tower.
- Louvre: Tons of art, like that Mona Lisa grin.
- Montmartre: Cute streets with killer crepes.

The "pal" vibe makes it friendly and focused. The whitepaper says try roles like "coach" for a pep talk.

## 5. Thinking Out Loud (Chain of Thought)

This is like making the AI explain itself. It's awesome for stuff like math, where it can trip up otherwise.

**Here's one:** A quick age puzzle.

Figure this out step by step: When I was 5, my sister was twice my age. Now I'm 20. How old is my sister?

Let's break this down step by step:

**When you were 5**, your sister was **twice your age**, which means:

$$\text{Sister's age then} = 2 \times 5 = 10$$

So, the age **difference** between you and your sister is:

$$10 - 5 = 5 \text{ years}$$

**Now you're 20**, so your sister must be:

$$20 + 5 = 25$$

Final Answer: Your sister is 25 years old.

The steps catch dumb mistakes. The whitepaper says it's a bit wordy, so don't overdo it.

## Messing with the AI's Vibe

Prompts are only part of it. The whitepaper talks about dials you can twist, like **temperature** and **word limits**.

- **Temperature:** Low (0.1) makes the AI boring but safe. High (0.8) gets weird and creative.
- **Word Limit:** Keeps the AI from rambling. Set it low for short answers.
- **Other Stuff (Top-K/P):** Picks likely words. Don't sweat these yet - defaults work fine.

**Check this:** Same question, different feels.

Tell me a quick story about a cat.  
Vibe: Wild, up to 15 words

Cat zoomed after a laser, flipping over couches like a furry ninja.

Same thing, Vibe: Chill, up to 8 words

Cat napped in a sunny spot.

**Why it's neat:** Wild vibe's fun; chill's to the point. The whitepaper says start with 0.2 for a nice middle ground.

## Best Practices for Prompt Engineering

Writing tech stuff's taught me to keep things clear. Here's what I've picked up from the whitepaper and screwing around myself:

- **Don't Make It Hard:** If your prompt's a mess, the AI's lost. Say "three Paris cafes" instead of "fun French hangouts."
- **Show What You Want:** Examples are like a map for the AI.
- **Be Picky:** Say "list it like this" to avoid surprises.
- **Guide, Don't Nag:** "Write a short story" beats "don't write a novel."
- **Jot It Down:** Note what you tried. I use a little chart to remember what worked.

## Using AI to Code: Saved My Butt

Last month, I had a folder stuffed with files I needed to rename. Doing it by hand? No thanks. The whitepaper shows how AI can spit out code, so I gave it a shot.

A script to rename files.

Make a Python thing that asks for a folder and slaps "old\_" on every file name.  
Vibe: Safe, plenty of words

### Output:

```
import os

folder = input("What's the folder? ").strip()

if not os.path.isdir(folder):
    print("Folder's not real.")
    quit()

for file in os.listdir(folder):
    old_path = os.path.join(folder, file)
    new_name = "old_" + file
    new_path = os.path.join(folder, new_name)

    # Skip directories if you only want to rename files
    if os.path.isfile(old_path):
        os.rename(old_path, new_path)

print("Done renaming!")
```

**Why it's neat:** Worked like a charm when I ran it. The whitepaper says double-check AI code,

though - it can goof up sometimes.

## Challenges and Fixes

AI's not perfect. Here's what I've hit, plus what the whitepaper says:

- **Fuzzy Asks:** "Talk about stars" might get you Hollywood, not space. Be clear.
- **Made-Up Junk:** AI might say "cats fly." Ask for facts or lists to stay real.
- **Stuck on Repeat:** It'll go "yep yep yep" if you're not careful. Shorten the word limit.

---

## Other mentions by Author

- [medium.com](#) | Published in Coding Nexus
- [medium.com](#) | Written by Algo Insights