Week 3 · Lesson 4 — Practical Exercise

Prompt Engineering With 5 Building Blocks (Zero-Shot & Few-Shot)

What you will accomplish

You will design and test two prompts for the **same task** using the five building blocks—**persona, instruction, context, exemplars, output constraints**. First, you will write a **zero-shot** prompt (no exemplars). Then, you will write a **short few-shot** prompt (2–3 exemplars). You will compare outputs on the **same inputs** and reflect on quality, controllability, and consistency.

Work in Google Colab or your preferred chat interface. Keep everything in one notebook/file and submit both prompts, model settings, raw outputs, and your reflection.

Task

Customer Review Triage — Given a single product review, the model must:

1. classify overall **sentiment** ∈ {positive, neutral, negative},
2. extract **aspects** mentioned and their **polarity** (positive/negative/unclear),
3. write a one-sentence **summary**,
4. suggest an **action** ∈ {none, support, refund, exchange}.

You may adapt wording, but keep the **same fields** so your zero-shot and few-shot results are comparable.

Inputs to test (use these three)

1. "Battery dies after 2 hours and support never replied. I'm really disappointed."
2. "Surprisingly fast shipping; setup was painless and the screen is gorgeous."
3. "Keyboard feels okay, not great. For the price it's fine, but I expected quieter keys."

A. Build your zero-shot prompt (no exemplars)

Include the 5 building blocks

1. **Persona (who is the model?)**
   "You are a senior NLP analyst at a retail electronics company. Your job is to triage customer reviews for swift action."
2. **Instruction (what to do?)**
   "Classify sentiment, extract aspects with polarity, summarize in one sentence, and recommend an action."
3. **Context (definitions, scope, style rules)**

- Sentiment: overall stance of the author.
- Aspect = product attribute (e.g., battery, screen, shipping).
- Action policy (simple): negative + hardware failure → refund/exchange; negative + service issue → support; otherwise **none** unless the text asks for help.

4. **Exemplars**
   Not included in zero-shot.
5. **Output constraints (format/schema, tone, length)**

- Return **only** valid JSON that matches the schema below—no extra text.
- Use lowercase labels for class fields.
- Keep the summary ≤ 25 words.

B. Build your short few-shot prompt (add 2–3 exemplars)

Reuse the same **persona, instruction, context, and output constraints**, but now add **concise exemplars** that demonstrate the mapping from review → JSON. Keep exemplars short and varied.

C. Compare and reflect (add to the end of your notebook/file)

What to report

1. **Prompts**: paste both full prompts (zero-shot & few-shot).
2. **Model & settings**: name (e.g., "gpt-3.5/4, Llama-X, etc."), temperature/top-p (or default).
3. **Outputs**: show JSON results for all three inputs under each prompt version.
4. **Comparison (≈150–200 words)**:
   - Where did **few-shot** improve consistency or field coverage versus zero-shot?
   - Any **hallucinated** aspects or wrong actions? Provide one concrete example.
   - Which block (persona, instruction, context, exemplars, constraints) most affected quality?
   - One **revision** you would make to harden the prompt (e.g., stricter schema, validation rule, tie-breaking for mixed sentiment).

Tips & pitfalls

- Keep **constraints** explicit (schema, label sets, length). This is where zero-shot often fails.
- In few-shot, prefer **small, atomic exemplars** that match your target style and edge cases.
- If models emit extra text, prepend: “Return ONLY JSON. No prose.” and set a **system** message when available.
- For robustness, add a fallback rule in context (e.g., *if no aspects are found, return an empty list*).

This exercise demonstrates how the five building blocks interact and why even **two or three** well-chosen exemplars can materially improve structure and fidelity.