Sneha Lokesh
Slokesh@clarku.edu

A3-Prompt_Engineering_Lab

Sneha Lokesh
Slokesh@clarku.edu

1) Introduction

Fast engineering is a careful design of entry practice to correct production from a model. While large models can be flexible for vague signals, small local models (eg Lama 3.2 1b) are greatly served with clear instructions, structured formats and work examples.

Goal. This laboratory evaluates four strategies-basic, structured production, a few shooting and chain-of-tree (COT)-a small, local runner model via Olama. We test realistic product features (ACME Coffee App Reference) and compare accuracy, perfection and clarity to get practical recommendations.

Contribution.

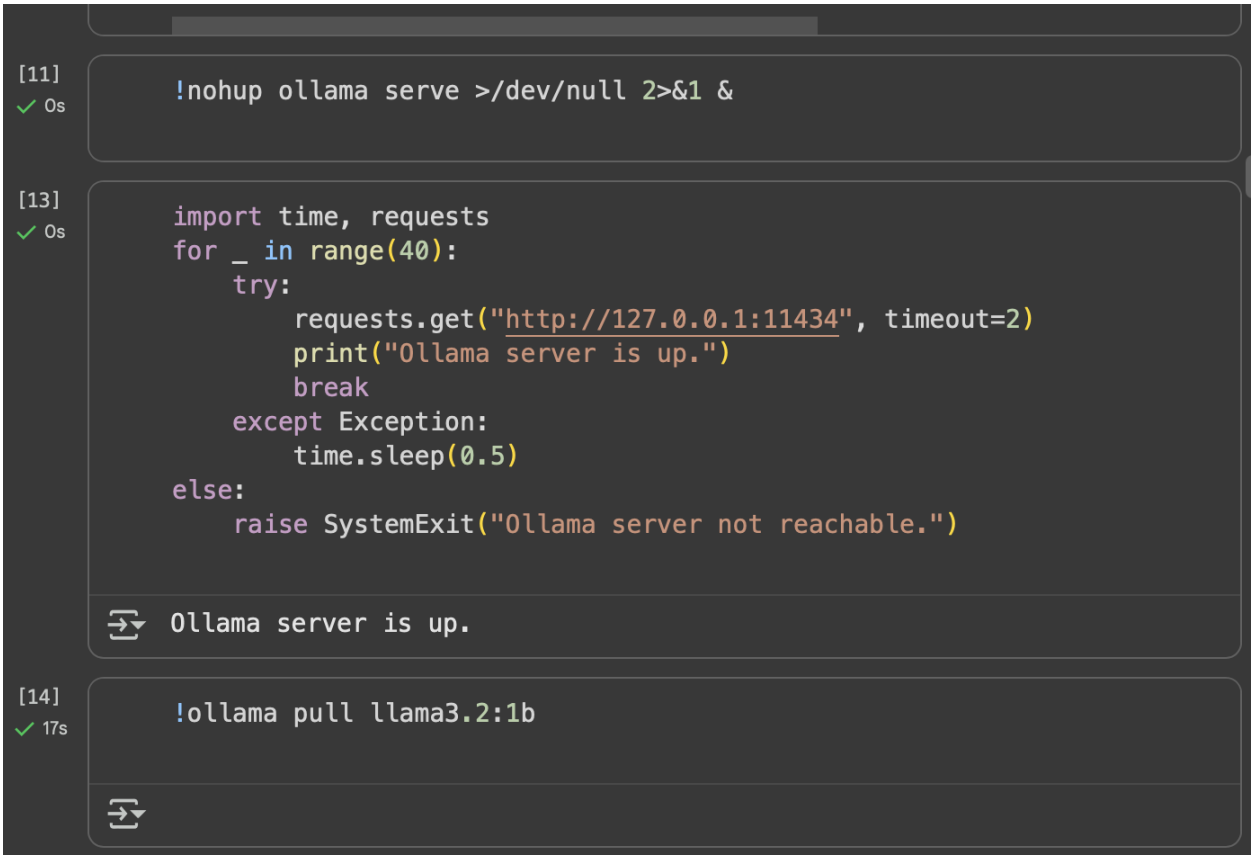
- A systematic, reproducible notebook running four early strategies.
- Quantitative and qualitative comparison (manual 0-5 ranking + light car hyuristics).
- Action -rich guidelines for using each strategy.

2) Model Setup & Methodology

Environment. Local/Colab runtime with **Ollama** serving **llama3.2:1b**.

- **Model:** LLaMA 3.2 (1B) via Ollama
- **Generation options (typical):** temperature≈0.2, optional seed=42, num_ctx=2048
- **Artifacts:** Stored under a3_outputs/step*.json plus ratings_template.csv

Sneha Lokesh
Slokesh@clarku.edu



Context document.
Acme Coffee launches a mobile app with customization, favorite, pickup planning, seasonal drinks, prices, referrals and a barista assistant. KPI includes DAU, AOV and D7 storage.

Tasks (eg).

- T1: Summarizes key feature
- T2: Lists KPI and explains why they matter
- T3: Proposes the experiment to increase AOV
- T4: Writes an FAQ to answer on redeeming reward

Evaluation.

- **Manual rating (0–5):** Correctness, Completeness, Clarity (filled in ratings_template.csv).
- **Auto heuristics (0–1):**

- *Structure* (valid JSON when required)
- *Grounding* (term overlap with context)
- *Brevity* (length penalty beyond ~80–160 words)
- *Format obedience* (e.g., CoT uses a “Final Answer:” line)

3) Experiment

3.1 Step 1 Basic Prompting

Prompt style. Direct questions and answers in context. No special format.

Why. Install baseline behavior and error mode (verbosity, lack of structure).

Example Prompt (T1).

“You are assistant assistant.

Reference: [Acme Coffee Reference]

Question: Combine the most important features of the Acme Coffee app. „

Expected behavior. Short paragraphs/tablets; May be right but can be verbs or incompatible.

3.2 Step 2 Structured Output Prompts

Fast style. Return only hard json, eg::

{"Answer": "...", "Supporting_facts": ["..."], "trust": 0.0}

Why. Improves reliability and analysis (important for pipelines).

Example Prompt (T2).

"Use keys answers, costing_facts, only with valid Json. Use only reference. If you put unknown, empty facts and confidence. 0. 0."

Overview. Structured signals reduced the ambiguity and corrected the evaluation (eg easy to automatically check the confidence/facts).

Sneha Lokesh
Slokesh@clarku.edu

3.3 Step 3 Few-shot Prompting

Fast style. 1-3 presented examples that match the desired style (eg, short QA or bullet features), and then ask new questions.

Why. Small models learn formatting and material expectations from examples (stylistic and semantic adjustment).

Example of some shot heading.

Season drinks with short pills about Q/A

Overview. Stability and clear lift in tone; Additional prose may still be involved without strict formatting of obstacles.

3.4 Step 4-Baret-AV-Thot (Children's Bed)

Fast style. "Think step by step ... end with the last answer: ...".

Why. Encourages clear arguments, which can improve multi -step tasks (planning, trade -offs).

Example Prompt (T3).

“Let's summarize the steps from the steps, and then give a short final answer. Overview. Best for the logic of quality and openness; Sometimes verbose ("short" was reduced with obstacles and a final north line).

4) Results

4.1 Manual Ratings (0–5)

- Average (purity, perfection, clarity) Average by strategy after filling A3_OutPuts/Ratings_template.csv.
- [Screenshot 7: A3_Outputs/Bardiagram from Manual_Prating_bar.png]
- Sample interpretation (replace with your actual numbers):
- Structured and some shots often lead to clarity and perfection to the responses from the fee style. COT often leads to purity of logic/experiment (T3). The original is acceptable for easy summaries, but less consistent.

Sneha Lokesh
Slokesh@clarku.edu

4.2 Automatic Heuristics (0–1)

- Calculation per step in all reactions: structure, grounding, briefness, format.
- [Screenshot Insest 3: A3_Outputs/Auto_heuristics_bar.png to Bardiagram]
- Sample interpretation:
- Structure: Quiet to -1.0, JSON SAMPS FOR STRUCTIONS. Grounding: High when reusable app/KPI terms; Useful signs against hallucinations. Brevity: Structured/some shot is short; The cot can be long. Format: Both cot and structured Json with "Final answer:".

5) Analysis & Discussion

Effect of structure. Strict Json improved the reliability and grading of ease (easy to pass, compare and shop). Smaller risks: The model sometimes adds extra prose - fixed by "only Json" instructions and verification.

Effects of examples (some shot). Get a more consistent answer without any-shot adjustment style and materials, heavy formatting barriers. Especially effective for "list features / suggestions experiments".

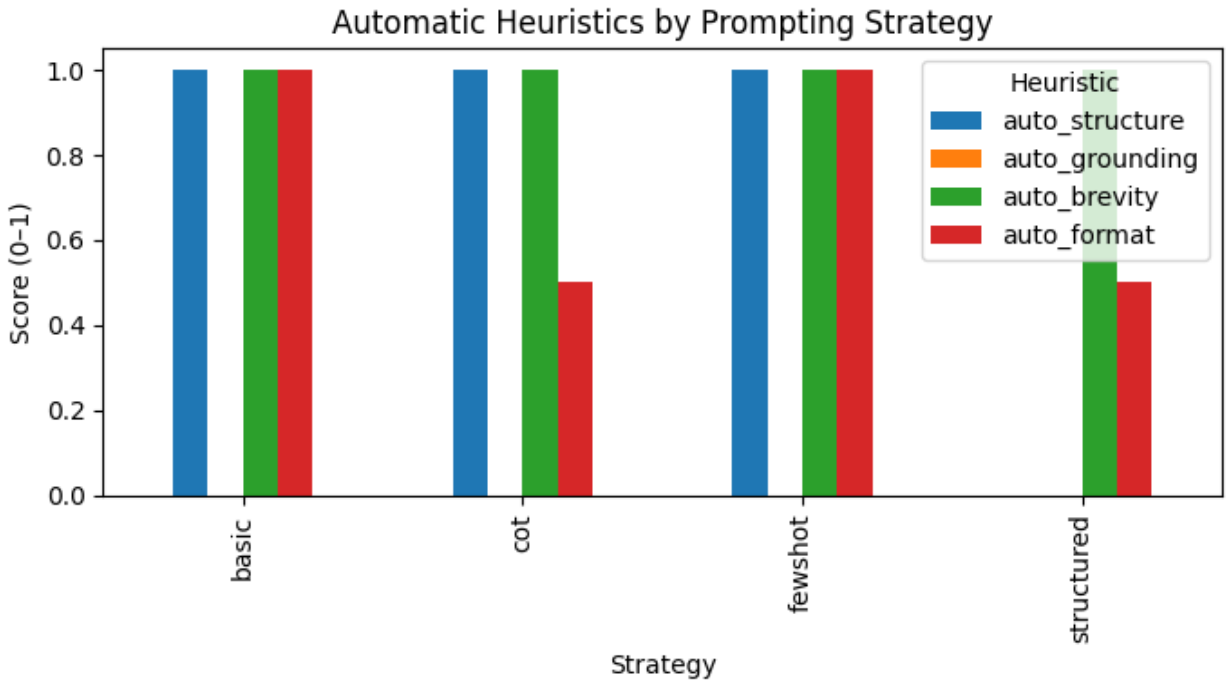
Effect of cot. COT improves logical quality, supports the A/B testing of suggestions and better justification for KPI explanation. It can be proper; Limitations include the capsule phase and require a brief "final answer".

Small model realities. The 1B model benefits from clear obstacles (format, concise, reference level grounding). Low temperature (.2) Stable output.

Challenges.

Topic Json operation ("only Json" + determined by the verification).

Without the need for a last row, the cot can end without a sharp answer. The final line's rules solved it. Grounding is improved when we only remind "only by using reference" and punishing inability to claims.



6) Recommendations

- Constitutes outputs for automation and grading of pipelines (dashboard, ETL).
- Some shots when tasks follow a pattern (FAQ, short bullet, marketing copy).
- Cot for plan/experiment/trade-off analysis (then take out one-line "final answer").
- Basic signs for fast, low road; Otherwise, light structure prefers.
- Keep the temperature low for stability; Add seeds to the copy of the copy.
- Add Confirmation: Check Json; Search for "Final Answer:" In the cot; Measure the grounding overlap.

7) Conclusion

The early strategy has a physical effect on the small model performance. In this laboratory, structured and some shoot improved clarity and stability, while COT improved logical quality. Choosing the right strategy depends on the task: Pipeline → Structured, pattern Qa → A small shot, regional/planning → cot. Apparently instructions, reference anchoring and simple assessment hooks (JSON verification, final response recovery) make small models practical for real projects.

8) Appendix

A. Prompt Templates (concise)

- **Basic:** direct Q/A with context.
- **Structured (JSON only):** enforce {"answer": "...","supporting_facts": [...],"confidence": 0.0}.
- **Few-shot:** 1–3 examples → "Now answer using the context below…".
- **CoT:** "Think step by step briefly… End with: Final Answer: <answer>".
- **[Insert small prompt snippets from your notebook for each step]**

B. Repro Instructions (local)

- Install Ollama; run ollama serve.
- ollama pull llama3.2:1b.
- Run the notebook cells in order; outputs saved to a3_outputs/.

C. Sample Output Artifacts

- a3_outputs/step1_basic.json
- a3_outputs/step2_structured.json
- a3_outputs/step3_fewshot.json
- a3_outputs/step4_cot.json
- a3_outputs/ratings_template.csv (filled by you)
- Charts: manual_ratings_bar.png, auto_heuristics_bar.png

Sneha Lokesh
Slokesh@clarku.edu