MAYO CLINIC

PROMPT ENGINEERING & AGENT SKILLS

FROM HAND-TUNED PROMPTS TO REUSABLE CLINICAL SKILLS

Muhammad Ali Khan

BEACON | Riaz Lab · Mayo Clinic

UIUC Medical-AI Workshop · Part 2 · 2026

THE STORY IN FOUR MOVES

From hand-tuned prompts to an agent that runs the whole task

1

Before

What prompt engineering used to take: expert, manual, repeated work

2

After

Skills & agents package the validated recipe once

3

Without the expertise

A non-expert invokes the skill and gets expert-level output

4

Tools

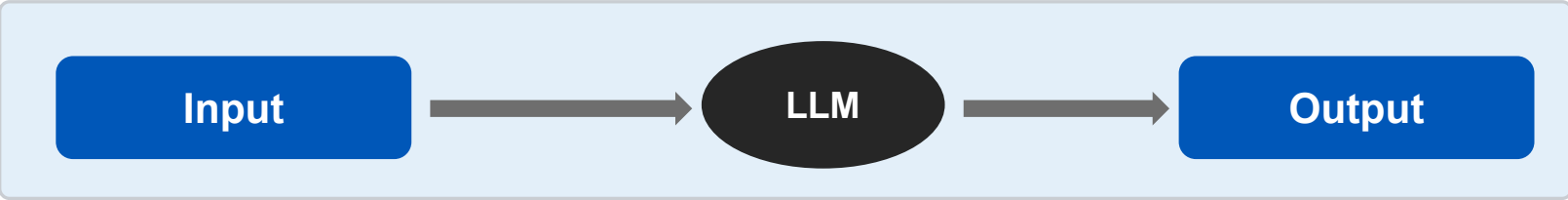
The agent calls tools to run the whole task; you still validate

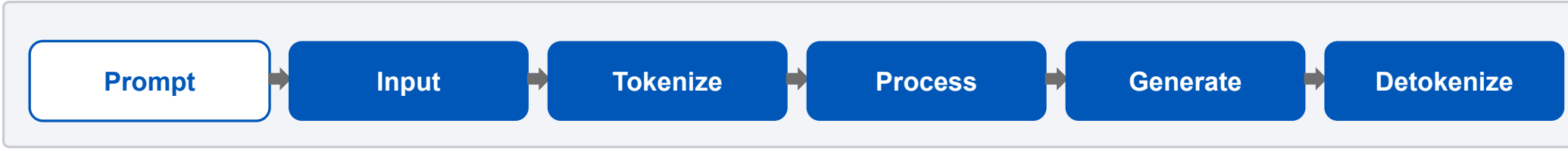
1 BEFORE: THE PROMPT-ENGINEERING PARADIGM

INFERENCE

An LLM only knows what you put in the prompt

Inference = using a pre-trained model to generate an output from new input. The weights are frozen; the only thing you control is the input.





Mental model: an extremely well-read resident who has read the whole internet, but has no access to THIS patient's chart unless you paste it in.

WHAT A PROMPT IS

An instruction plus the report, with five traits that make it effective

A prompt = a query (what you want) with or without an input text (the report), given to the model to elicit a specific output.

Clear & specific
say exactly what to decide

Sufficient context
give the report + the rules

Structured format
name the output shape

Iteratively refined
tune it; rarely right first try

Decomposed
one ask at a time when it matters

BUILDING A PROMPT

One prompt = the query + the input text

Input

Query

What is the cancer status in this radiology report?

Impression from a radiology report

Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

Prompt

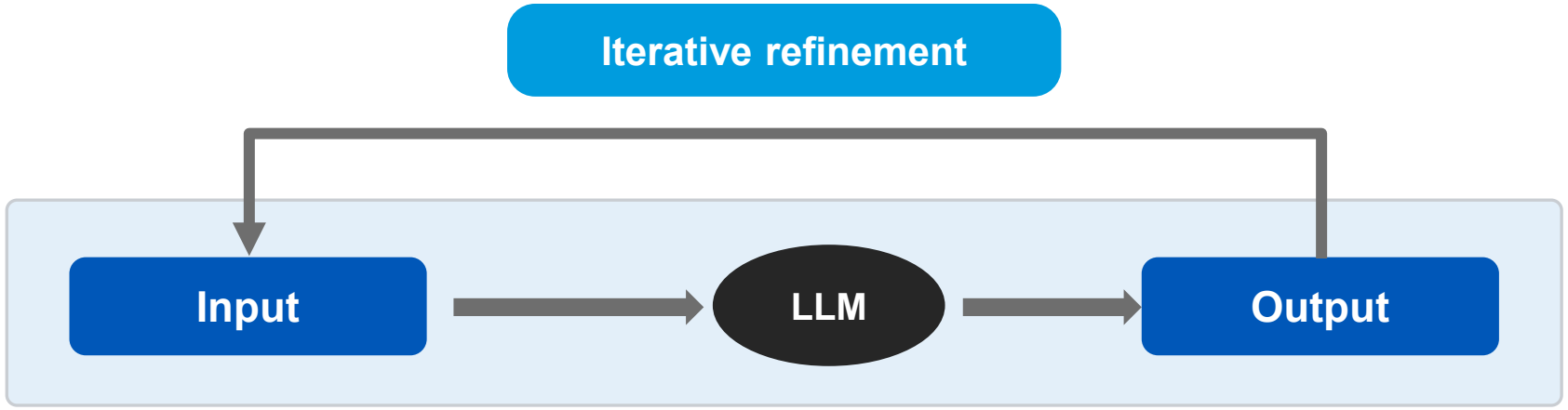
What is the cancer status in this radiology report?

Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

PROMPT ENGINEERING

We refine the prompt, not the model

Prompt engineering = designing and refining the prompt to get the output you want. It is the part of the system you control.



The loop: send a prompt, read the output, change ONE thing, send again. We can't retrain a large model, and we don't need to.

REFINEMENT LEVERS

A small menu of standard moves

Shape the ask

Add structure: bullets / sections, not prose

Specify the output format: Yes/No, a table, JSON

Decompose: one variable at a time

Add a role: “You are an expert pathologist…”

Add constraints: “answer only from the report”

Show, reason, vote

Give examples (few-shot): 1–3 worked demos

Ask for reasoning (CoT): “think step by step”

Vote across runs (self-consistency): take the majority

Use these when accuracy or format matters.

Example: iterations 1–4 use the same impression, including one change that backfires.

REFINING: ITERATION 1

Accurate prose, but you can't parse it

Prompt

What is the cancer status in this radiology report?

Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

GPT-4o **Response**

The radiology report indicates that the cancer has worsened. There is an increase in overall metastasis, with more and larger hepatic metastases, increased retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. A significant hepatic metastasis in the right lobe is noted. There is also a new small right lower lobe consolidation, which is likely aspiration pneumonia.

What we learn

✓ Clinically accurate

Even calls the lung finding aspiration pneumonia, correctly

⚠ Not parseable

It's a paragraph: no database column, no count

REFINING: ITERATION 2

A sharper question introduced a clinical error

Prompt

Identify if there is evidence of disease progression in the following radiology impression.

Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

GPT-4o **Response**

Yes, there is evidence of disease progression:

- **Hepatic metastasis:** increase in size and number.
- **Retroperitoneal nodes:** increasing, worsening disease.

✗ **Right lower lobe:** "new consolidation indicating worsening disease."

Wrong: the report says 'likely aspiration pneumonia', not cancer.

⚠ The lesson: validate every change

✓ Structure improved

Organized, bulleted

✗ New clinical error

Misread aspiration pneumonia as progression

Key takeaway: every change can fix one thing and break another. Re-validate each change against the report.

REFINING: ITERATION 3

Naming the task + format yields clean Yes/No

Prompt

TASK: Identify (1) any evidence of cancer, (2) progression/worsening, (3) response/improvement.

INSTRUCTIONS: For each, answer Yes or No. No explanations.

IMPRESSION: Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

GPT-4o **Response**

Any evidence of cancer: **Yes**

Progression / worsening: **Yes**

Response / improvement: **No**

Why this works

✓ Clear task
3 named outcomes

✓ Precise format
Yes/No only

✓ Parseable
Two lines of code load it

Note: structure makes errors easier to catch and audit; it doesn't guarantee they're correct. You still validate.

REFINING: ITERATION 4

One output-format line buys structure + reasoning

Prompt

TASK: Identify (1) cancer, (2) progression, (3) improvement.

INSTRUCTIONS: Answer Yes/No with a short explanation.

OUTPUT FORMAT: `<Outcome: Yes/No; Explanation: X>`

IMPRESSION: Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

GPT-4o **Response**

Evidence of cancer: Yes
Hepatic and nodal metastases are definitive signs of cancer.

Progression / worsening: Yes
Increase in size and number of lesions → progression.

Response / improvement: No
No reduction in size, number or extent of metastases.

The sweet spot for clinical use

✓ Structured + explained
Fixed shape, with reasoning

✓ Auditable
A human can see WHY

✓ Parseable
Splits cleanly downstream

ITERATION 1 VS 4

Four small edits turned prose into structured data

Iteration 1

The cancer has worsened… an increase in overall metastasis, larger hepatic metastases, increased lymphadenopathy, sclerotic osseous metastasis, and a new right-lower-lobe consolidation, likely aspiration pneumonia.

Iteration 4

Evidence of cancer: Yes, definitive metastatic signs.
Progression: Yes, increasing size & number.
Improvement: No, no reduction in disease.

Only the prompt changed: no new model, no training, no extra code

1

Unstructured
narrative, hard to parse

→

4 validated edits
structure · task · format

4

Ready to analyze
structured, explained, parseable

FEW-SHOT LEARNING

Models learn a new task from a few in-prompt examples

Zero-shot

Task description only. No retraining.

```
Translate English to French:
cheese =>
```

One-shot

Description + one example.

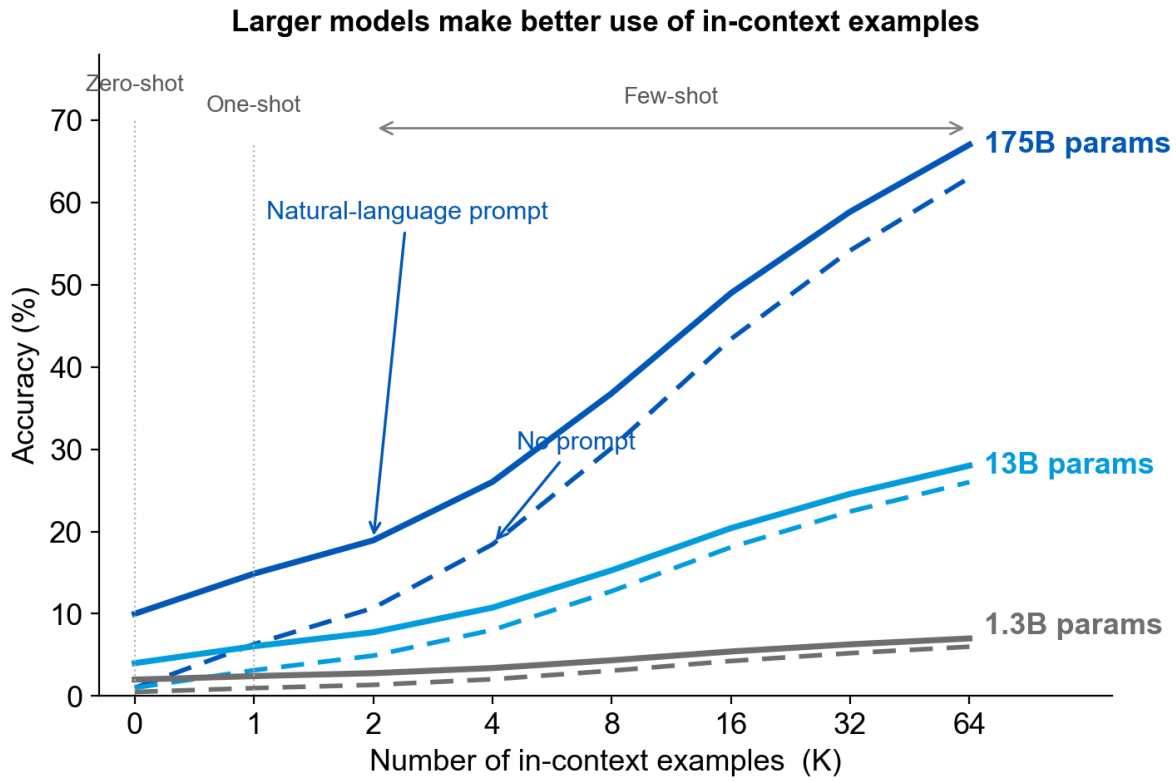
```
sea otter => loutre de mer
cheese =>
```

Few-shot

Description + a few examples.

```
sea otter => loutre de mer;
peppermint => menthe poivrée;
cheese =>
```

In the clinic → report snippet : desired output.

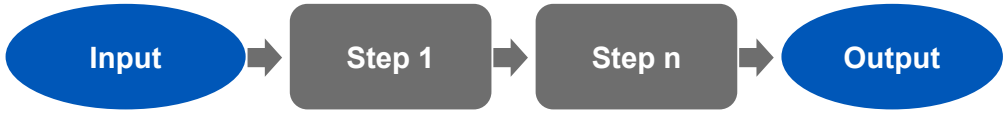


Schematic recreation for teaching; based on Brown et al. 2020 (GPT-3).

REASON & VOTE

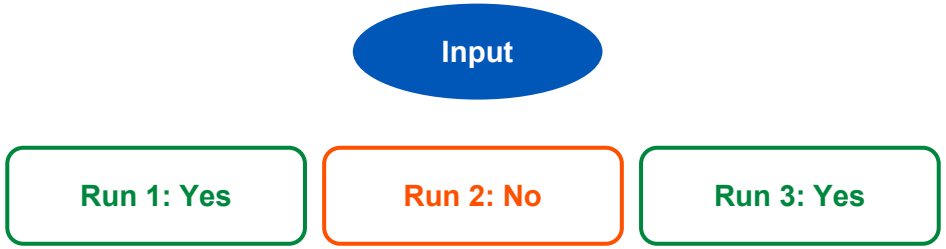
Chain-of-thought and self-consistency add reliability

Chain of thought



Ask the model to reason step-by-step before answering. This helps on multi-step problems such as staging. Trade-off: more tokens, and the reasoning is a rationale, not a guarantee.

Self-consistency



Run the same prompt several times; take the majority answer. This smooths out randomness, and disagreement is a useful flag for human review.

Framing: chain-of-thought = show your working; self-consistency = a second and third opinion. Both buy reliability at the cost of compute.

THE COST OF THIS PARADIGM

Prompting works, but it doesn't scale, reach tools, or transfer

Repeat it every time

The validated prompt lives in a notebook cell or your head. New variable, new report type, new teammate → you start over.

No tools

Text in, text out. It can't pull the report from the EHR, search a long document, look up a code, or run the voting loop. You do all of that by hand.

Needs a prompt-engineer

Reaching iteration-4 quality reliably takes format contracts, few-shot, chain-of-thought, self-consistency and validation discipline. That is expertise most clinicians don't have.

Skills and agents are built to remove exactly these three.

A fluent prompt can still be wrong, so validation stays part of the workflow.

2 AFTER: HOW SKILLS & AGENTS IMPROVED IT

WHAT A SKILL IS

It packages your best prompt, its tools, and its rules just once

An agent skill is a reusable capability that helps an agent do a specific task more reliably. It is the validated prompt, written down so it travels.

A skill bundles:

- Task-specific instructions (the prompt you engineered)
- Tools or helper functions
- Examples or templates
- Rules for when & how to use it
- The repeatable workflow

In our running example

the skill stores the validated **iteration-4 prompt**, plus a tool to **find the relevant findings**, so nobody re-engineers it.

But a consistent process can still be consistently wrong, so validate the skill before you depend on it.

SKILLS REMOVE THE COSTS

Each limit of prompting, cleared

The cost (prompting alone)	How a skill clears it
Repeat it every time	Write the validated prompt ONCE; reuse it across reports, variables and sites. You invoke it, you don't re-tune it.
No tools	The agent calls tools (search the report, look up a code, run the voting loop) instead of you doing it by hand.
Needs a prompt-engineer	The engineering is baked in, so it no longer has to live in every user's head (who that helps, next).

Same expert output, reproducibly, without re-doing the prompt engineering (you still validate the skill).

INSIDE A SKILL

Just a folder: workflow + tools + knowledge

```
report_analysis_skill/
|
|-- SKILL.md                    # the workflow
|
|-- scripts/
|   `-- evidence_finder.py      # the tool
|
`-- reference/                  # knowledge
    |-- cancer_terms.md
    |-- report_sections.md
    `-- output_examples.md
```

WORKFLOW

SKILL.md: the step-by-step process the agent follows

TOOL

scripts/: helper functions like evidence_finder.py

KNOWLEDGE

reference/: terms, report sections, examples

THE SKILL.MD FILE

The whole skill is one readable markdown file

```
---
name: report-analysis
description: Analyze clinical reports using report-based evidence.
---

# Report Analysis

When to use:  When the user asks a question about a clinical report.

Workflow:
  1. Read the user's task.
  2. Locate the most relevant report sections.
  3. Run scripts/evidence_finder.py to find supporting evidence.
  4. Answer based only on the report text.
  5. Follow the requested output format.

Output rule:  Return the answer in the requested format.
              Do not include unsupported clinical assumptions.
```


3 EFFECTIVE PROMPTS WITHOUT PROMPT-ENGINEERING EXPERTISE

PROMPTS WITHOUT EXPERTISE

A teammate invokes the skill and gets expert-level output

Without a skill

Every new user must know:

- name the task and the 3 outcomes
- specify a parseable output format
- add role + anti-hallucination guardrails
- know when to add few-shot / CoT / voting
- validate every change

…and redo it for every report type.

With the skill

The user writes:

“What's the cancer status in this report?”

The skill supplies the validated iteration-4 prompt, the guardrails, the tool, and the output format, all automatically.

An expert-quality prompt, no prompt-engineering expertise needed.

The clinician still applies judgement and validates the output.

The prompt-engineering expertise now lives in the skill, not in every user. *This mirrors how we write clinical protocols instead of re-deciding each time.*

PROMPT OR SKILL?

When a prompt is worth turning into a skill

A good prompt → **A reusable skill**

graduate it when…

- ✓ You'll reuse it: the same task recurs across reports, studies, or sites
- ✓ It needs tools: searching sections, looking up codes, calling a function
- ✓ Others should use it: a teammate should get the same result you do
- ✓ It needs validation: the output feeds a registry, dashboard, or decision
- ✓ It bundles knowledge: definitions and examples travel with it

4 AGENTS CALL TOOLS TO RUN THE TASK

THE AGENT DOES THE WORK

Same impression, now the agent runs the task

What the clinician does

Supplies the report and asks:

“Cancer status & progression?”

Increase in the overall metastasis in the form of increase in size and number of hepatic metastasis, increase in size of retroperitoneal lymphadenopathy, and sclerotic osseous metastasis. The large hepatic metastasis in the right hepatic lobe is seen. New small right lower lobe consolidation is likely aspiration pneumonia.

No engineered prompt, no manual tools.

What the skill + agent do

1 **Loads the validated prompt**

2 **Calls a tool** `find_relevant_findings()`
→ surfaces the metastasis lines; guardrails keep pneumonia out of progression

3 **Runs self-consistency ×3**
→ takes the majority answer

What comes back

cancer: Yes

progression: Yes

improvement: No

pneumonia not called progression

→ clinician reviews & signs off

The clinician did not hand-build the prompt or run the tools: the three costs are addressed.

The workflow specifies the steps, and the clinician still validates the result.

AGENTS CALL TOOLS

That was one tool; the same pattern generalizes

A prompt answers. An agent and skill run the task's steps (actions a prompt never could take), then hand you a structured result to review.

Retrieve	Search	Look up
pull the report from the EHR or a file store	find the relevant section of a long document	map a diagnosis to an ICD / SNOMED code, a drug to RxNorm
Compute	**Write**	**Hand off**
re-run the prompt N times and tally the majority	emit a structured row to a registry or dashboard	route the result to a clinician before it's used

Same skill, many report types: swap the question (cancer? Gleason? margins?) and the workflow and tools stay.

Tool use comes from the workflow, and the result still gets validated.

KEY TAKEAWAYS

What to carry out of this session

1

Before: prompting is expert, manual, repeated work

We refined one prompt over four steps to get structured, auditable output, and watched a 'better' prompt introduce a clinical error.

2

Skills package that work once

The validated prompt, tools, and rules become a reusable, reviewable capability, like a clinical SOP.

3

So non-experts get expert output

A teammate invokes the skill and gets the same result; the expertise lives in the skill, not in every user.

4

And agents call tools to execute

Retrieve, search, look up, compute, write: the agent runs the whole task. You still validate against ground truth.

REFERENCES & FURTHER LEARNING

- **Anthropic. Agent Skills (overview & authoring)**

  platform.claude.com/docs/en/agents-and-tools/agent-skills/overview

- **OpenAI. Function calling (tool use)**

  developers.openai.com/api/docs/guides/function-calling

- **Brown et al., 2020. "Language Models are Few-Shot Learners" (GPT-3)**

  arxiv.org/abs/2005.14165

- **Wei et al., 2022. Chain-of-Thought Prompting**

  arxiv.org/abs/2201.11903

- **Wang et al., 2022. Self-Consistency improves Chain-of-Thought**

  arxiv.org/abs/2203.11171

QUESTIONS & ANSWERS