Carnegie Mellon University

Lecture 2: Foundational Concepts & Prompt Engineering

SPRING 2026

MOHAMED FARAG

FARAG@CMU.EDU

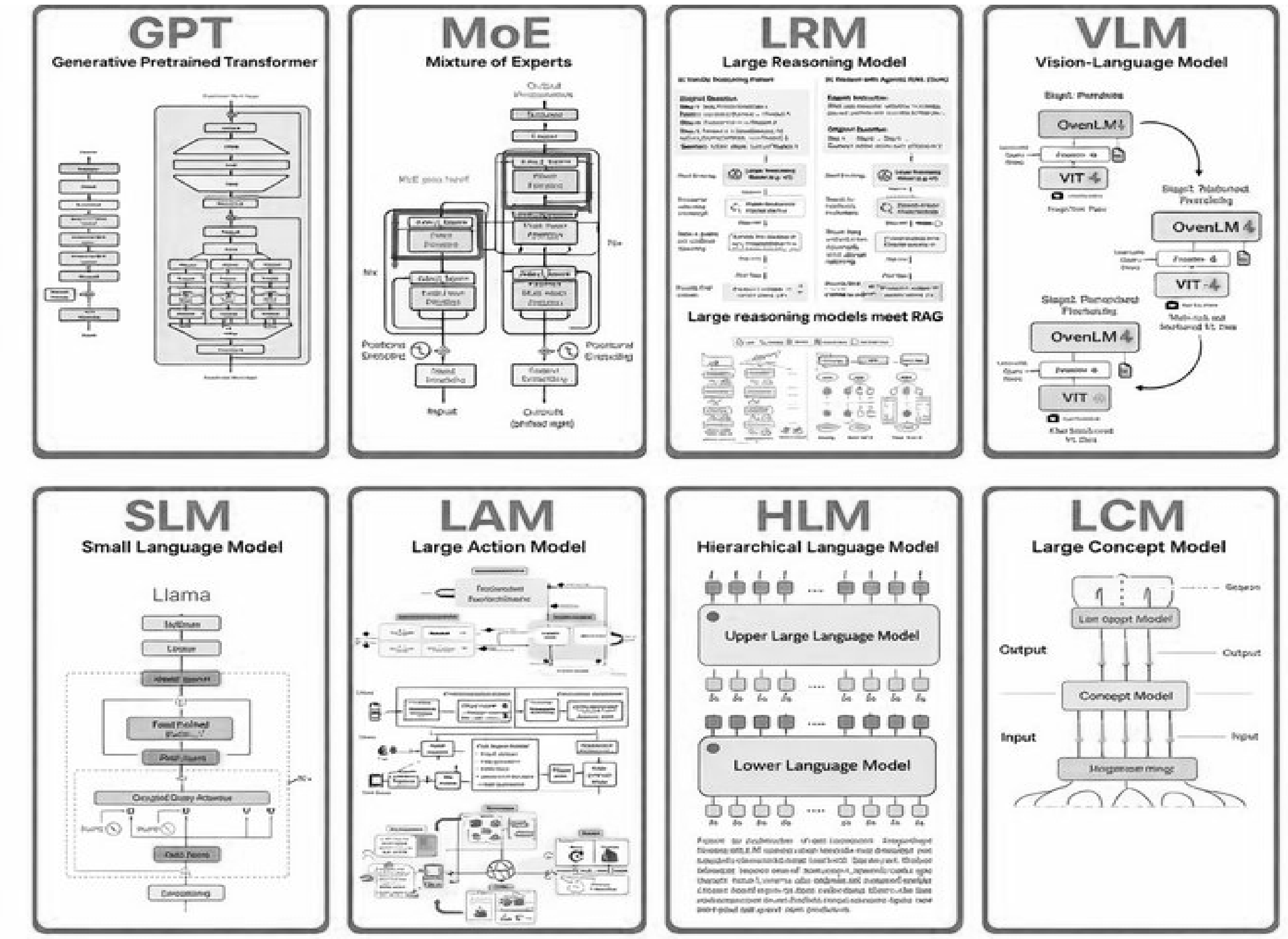
AGAI
Applied Generative AI

Agenda

- Model Architectures & Paradigms in Generative AI
- Exercises
- Prompt Engineering (Components and Strategies)
- Summary
- Readings

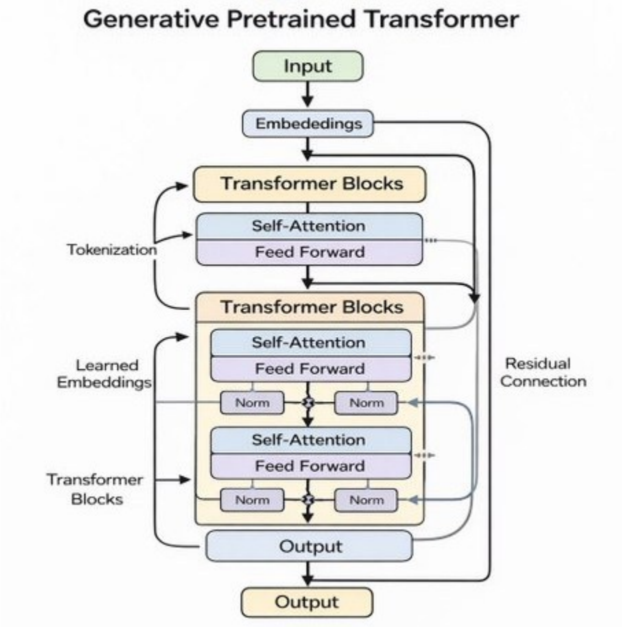
Are LLMs the only kind of models behind generative AI systems?

Model Architectures & Paradigms in Generative AI



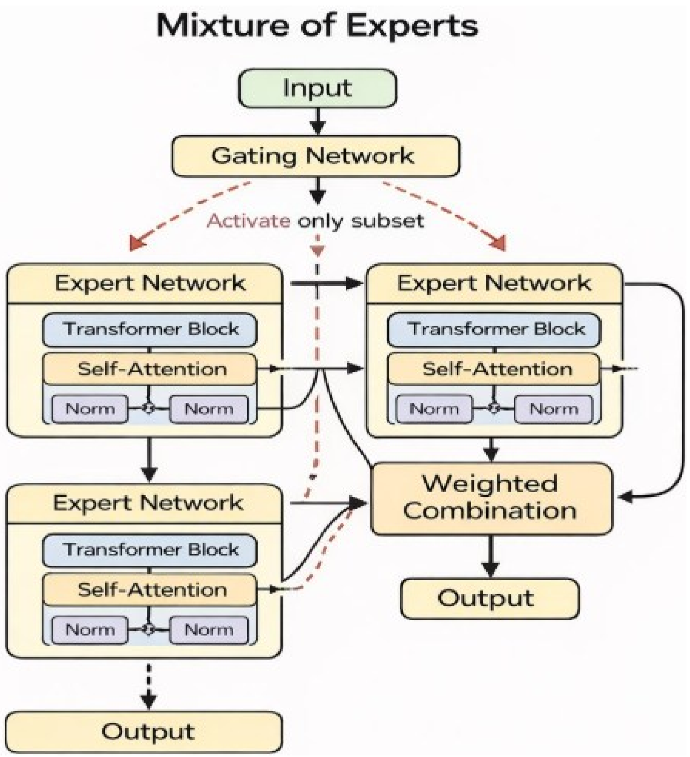
1. GPT-style LLMs (Dense Transformer Models)

- Dense transformer models are general-purpose language models optimized for natural language understanding and generation.
- Trained on massive datasets.
- Good for conversations and creative tasks.
- May struggle with complex mathematical tasks, visual tasks and action planning.
- Examples: GPT 4, LLaMA 2/3, Gemini text-only variants.



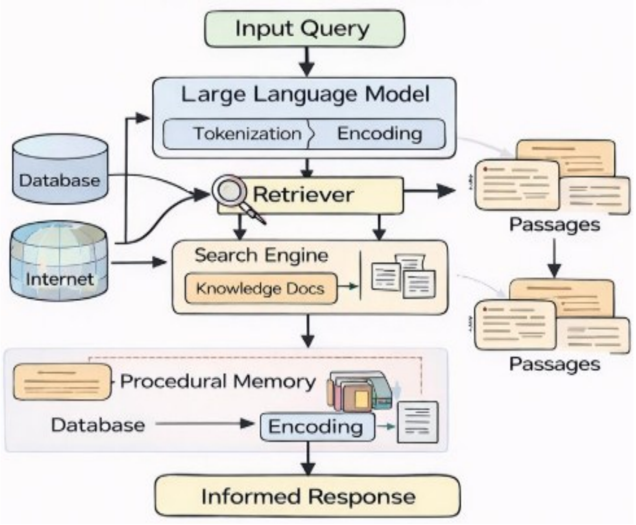
2. MoE Models

- MoEs replaces a single large model with multiple specialized expert models.
- A router (gating network) selects the appropriate expert(s) for each input.
- Good for 1) very large capacity with low inference cost environments and 2) Diverse tasks (code + math + reasoning).
- Avoid using them in resource-constrained environments or when you need uniform behavior.
- Examples: Mixtral 8×7B, MoE-LLaMA, Gemini 2.5 and 3.



3. Large Reasoning Models (LRMs)

- LRMs combine Chain-of-Thought reasoning with RAG-like capabilities.
- Good for tasks involving planning, deduction and strategy.
- Avoid using them for simple lookup tasks.
- Examples: GPT-o1, Gemini 2.5 and 3 Pro models.



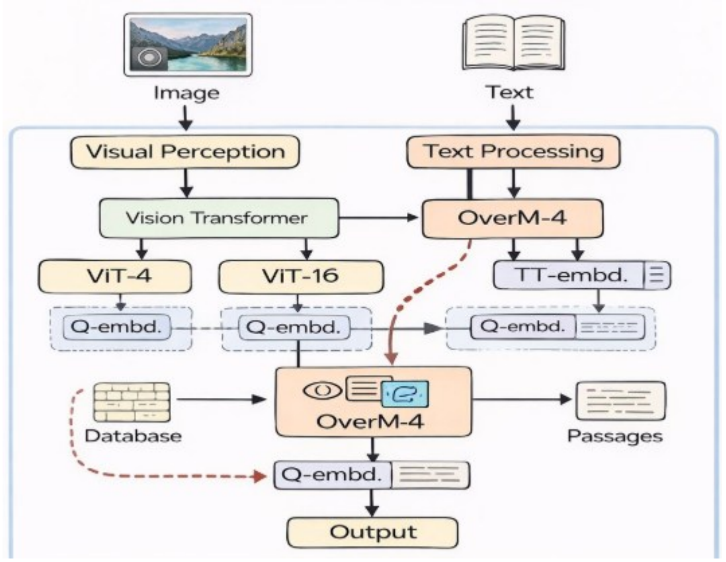
4. Vision Language Models (VLMs)

- VLMs combine visual inputs with textural inputs to generate multi-modal embeddings.

- Good for tasks involving document understanding, UI automation, image analysis and robotics perception.

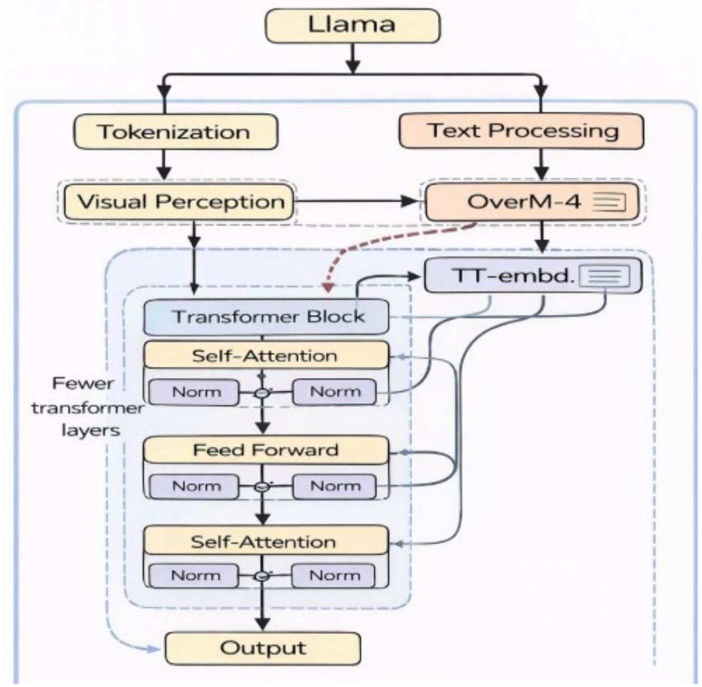
- Generally expensive and slow so avoid using them for pure text-based tasks.

- Examples: GPT-4V, Qwen 2.5-VL, LLaVA.



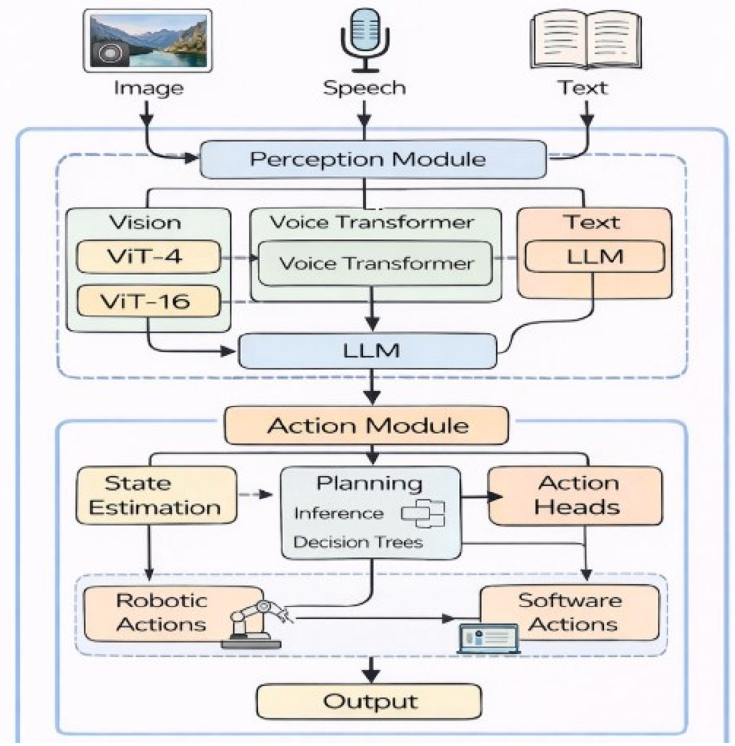
5. Small Language Models (SLMs)

- SLMs are tiny models (1B-7B parameters) optimized for specific tasks.
- SLMs are generally fast and have low inference costs.
- Good for edge-devices, on-prem systems and resource-constrained environments.
- Avoid using them individually for complex reasoning tasks.
- Examples: Phi-2, Phi-3, Gemma 2B, Gemma 7B.



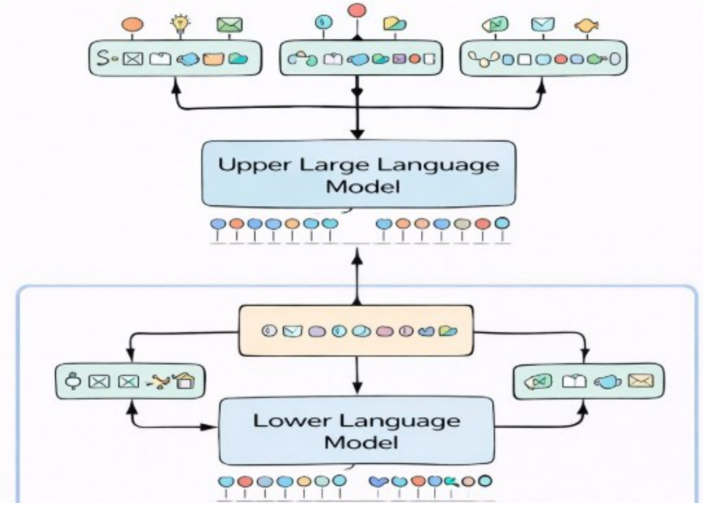
6. Large Action Models (LAMs)

- LAMs don’t just communicate with you. They take actions in real world.
- Used for Robotic control, workflow automation and heavy-resource agents.
- Avoid using them without safety guardrails. A mistake may lead to a potential hardware failure or life-critical situation.
- Examples: Adept ACT-1, Google PaLM-E.



7. Hierarchical Language Models (HLMs)

- HLMs are system-level orchestrations of language models.
- HLMs use two-tier architecture to handle complex tasks by decomposing them.
- Higher-tier is used for planning and task decomposition.
- Lower-tier is used for execution of individual tasks.
- Used for complex tasks that can be decomposed further.
- Generally slow and carries the compound bias of two LLMs.
- Examples: Hierarchical workflows in LangGraph, AutoGPT/BabyAGI.



8. Large Concept Models (LCMs)

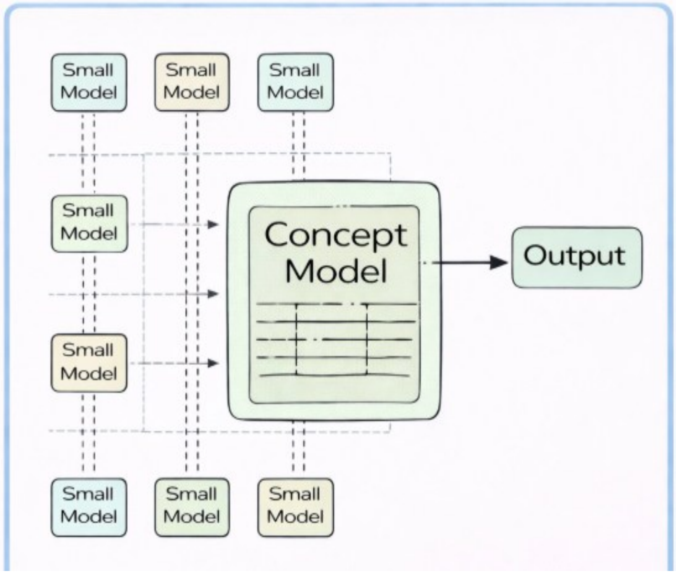
- LCMs operate over abstract concepts and structured knowledge representations rather than raw text.

- Analogy: LLMs think in tokens while LCMs think in ideas!

- Some researchers note that LCMs could be the future of LLMs. They are in the experimental phase.

- LCMs aim to reason at the conceptual level, not the word level.

- Example: Neuro-Symbolic Concept Learner (NSCL).



Q1. Which GenAI Model to use?

- Students are asked to build an assistant that helps users understand complex charts and diagrams in research papers (e.g., confusion matrices, system architecture diagrams).
- The user uploads an image of a chart and asks:
    - What does this graph show?
    - What conclusion can I draw from it?

Q2. Which GenAI Model to use?

- Students are building a local code review assistant that runs on a laptop or edge device to:
  - Flag style violations.
  - Detect simple bugs
  - Suggest minor refactors.
- No internet access is allowed.

Q3. Which GenAI Model to use?

- Students are designing a multi-domain tutoring system that supports:
  - Math problem solving.
  - Programming help.
  - Essay feedback.
  - Science explanations.
- Thousands of students may use the system simultaneously.

Prompting Engineering

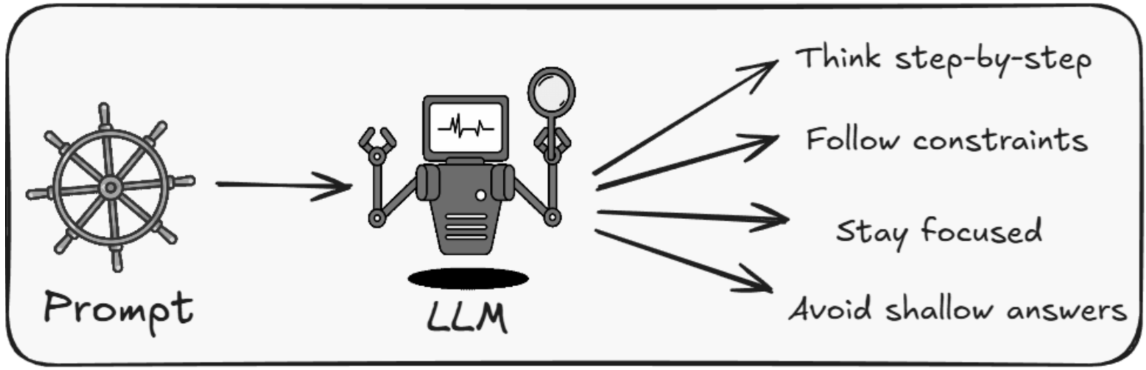
Reminder: LLM Limitations

LLMs are not perfect! Keep in mind the following limitations with LLMs:

- LLMs don’t work well with tabular data and mathematical calculations (Use supervised learning instead).
- Length of Input and Output of LLMs is limited, implying limited data to be provided and potentially limited output to be consumed.
- Knowledge Cutoff: LLMs are limited to the training date.
- Bias and Toxicity.
- Hallucinations!

Overcoming LLM Limitations: Prompt Engineering

- Prompt engineering refers to the systematic design of effective prompts for interacting with AI models.
- Prompt Engineering requires meticulously designing prompts to include essential context, guiding the language model towards more accurate and relevant responses without changing model weights or parameters.
- Bad Prompt = Bad Output.



AI Prompt Components: Required Elements

1. **Task:** What you want the AI to do (for example: write, summarize, or generate something).

2. **Format:** How the answer should look (length, type like an email or blog post, file type, etc.).

3. **Topic:** What the request is about? What is the main subject?

4. **Tone or style:** How it should sound (serious, casual, professional, friendly, etc.)?

5. **Context:** Extra background or details that help the AI understand the situation or audience.

6. **Requirements or constraints:** Rules the AI must follow, including what to include or avoid.

AI Prompt Components: Optional Elements

1. **Goal:** Tell the AI what outcome you want. what you want the reader to think, feel, or do.
   - For example, convince the reader or get them to take action.

2. **Role:** Tell the AI who it should act as or write from the perspective of.
   - For example, a teacher, an expert, a student, or a specific type of audience.

AI Prompt Example

Act as a professional career coach and write a short follow-up email about a recent job interview. The email should be professional and confident, no longer than 150 words, and written in a polite and positive tone. The context is that the interview happened two days ago, and the candidate wants to thank the interviewer and restate interest in the position. The goal is to encourage the hiring manager to move forward with the candidate. Do not mention salary or benefits, avoid sounding desperate, and end with a clear but polite call to action.

AI Prompt Example

Act as a professional career coach and write a short follow-up email about a recent job interview. The email should be professional and confident, no longer than 150 words, and written in a polite and positive tone. The context is that the interview happened two days ago, and the candidate wants to thank the interviewer and restate interest in the position. The goal is to encourage the hiring manager to move forward with the candidate. Do not mention salary or benefits, avoid sounding desperate, and end with a clear but polite call to action.

But .. This prompt is unstructured …
How would we optimize it?

AI Prompt Example using Structured Prompting

#Role: You are a professional career coach.

#Task: Write a short email.

#Topic: Following up after a job interview.

#Format: A professional email, no more than 150 words.

#Tone / Style: Polite, confident, and professional.

#Context: The candidate interviewed two days ago and wants to express appreciation and restate interest in the role.

#Goal: Encourage the hiring manager to move forward with the candidate.

#Requirements / Constraints:

- Do not sound desperate.
- Do not mention salary or benefits.
- End with a clear but polite call to action.

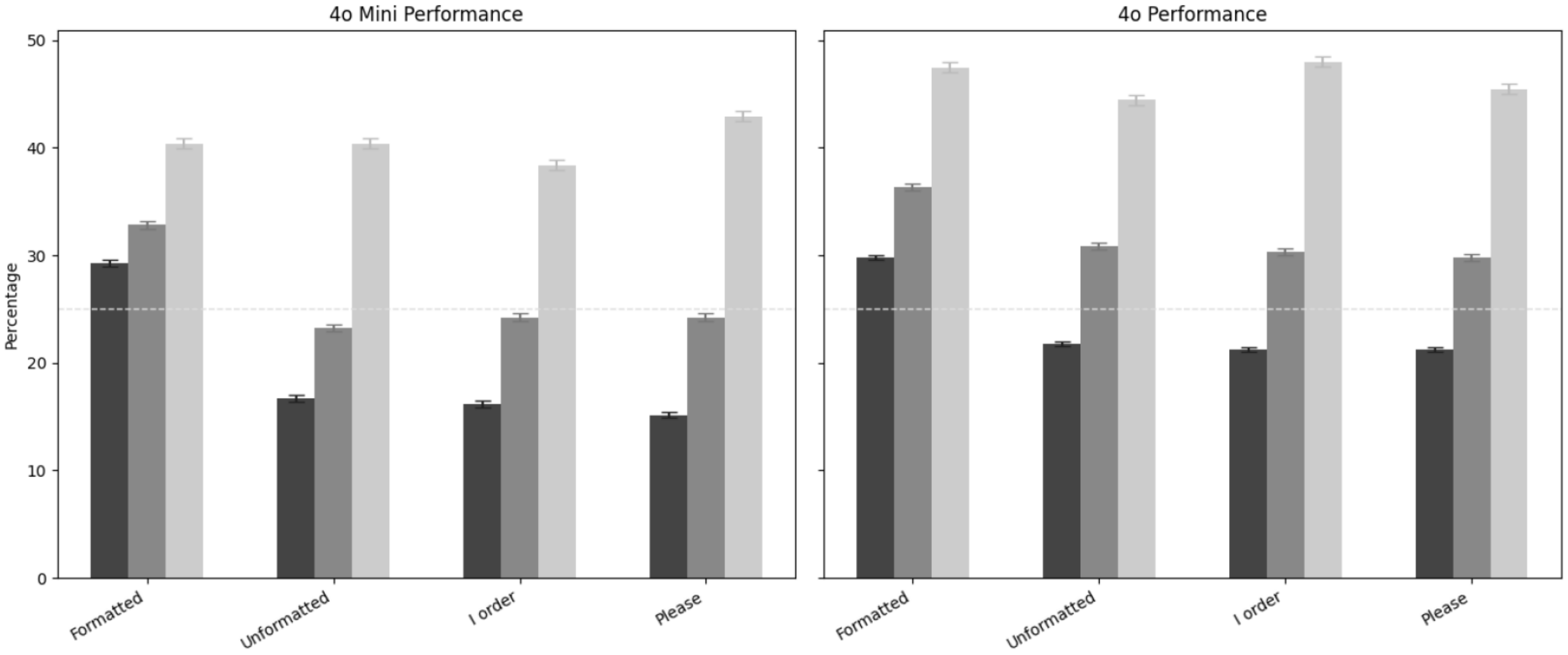
Is Structured Prompting Effective?

Research shows that using structured prompts can improve the LLM output accuracy by 10-20%. Refer to the following papers:

- https://arxiv.org/abs/2402.11770
- https://arxiv.org/abs/2305.06599
- https://arxiv.org/abs/2506.16123

Is Structured Prompting Effective? – Cont’d

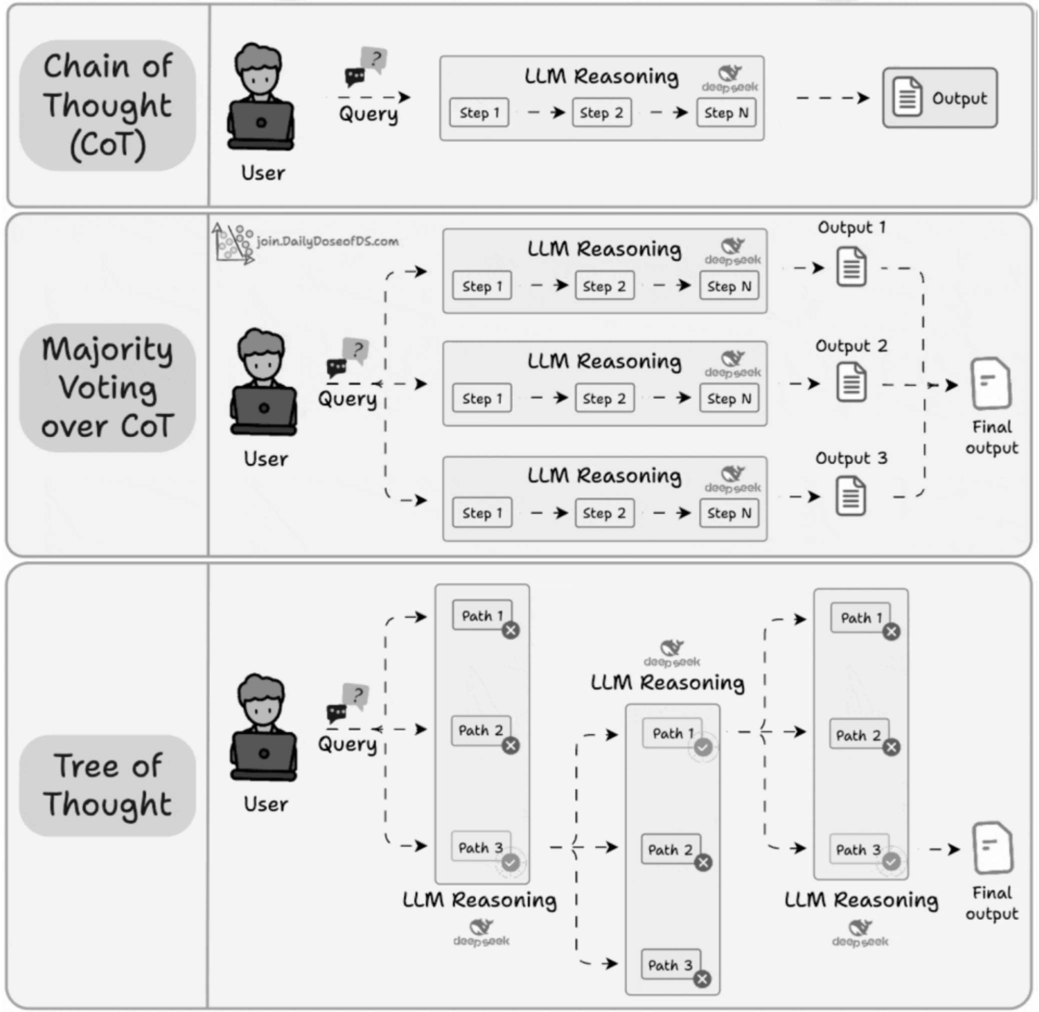
Researchers from University of Pennsylvania found that the structure and formatting of prompts can significantly affect LLM performance.
Specifically, well-structured prompts often outperform unformatted ones.



Using reference system prompt, temperature=0, and n=100 for each question.

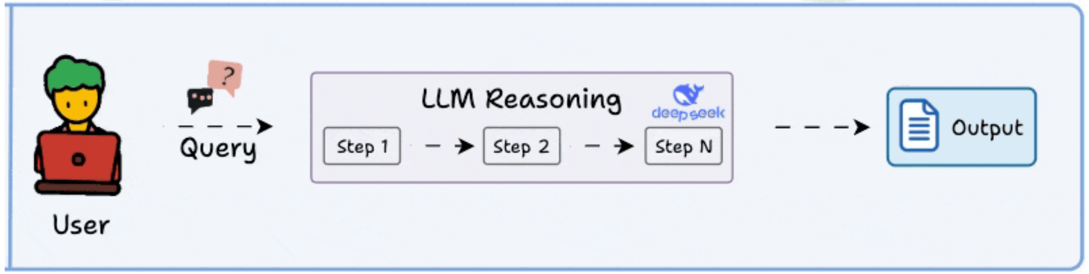
Figure 1 from Meincke, Lennart and Mollick, Ethan R. and Mollick, Lilach and Shapiro, Dan, Prompting Science Report 1: Prompt Engineering is Complicated and Contingent (March 04, 2025). Available at SSRN: https://ssrn.com/abstract=5165270

Prompting Strategies



Prompting Strategies: Chain of Thought (CoT)

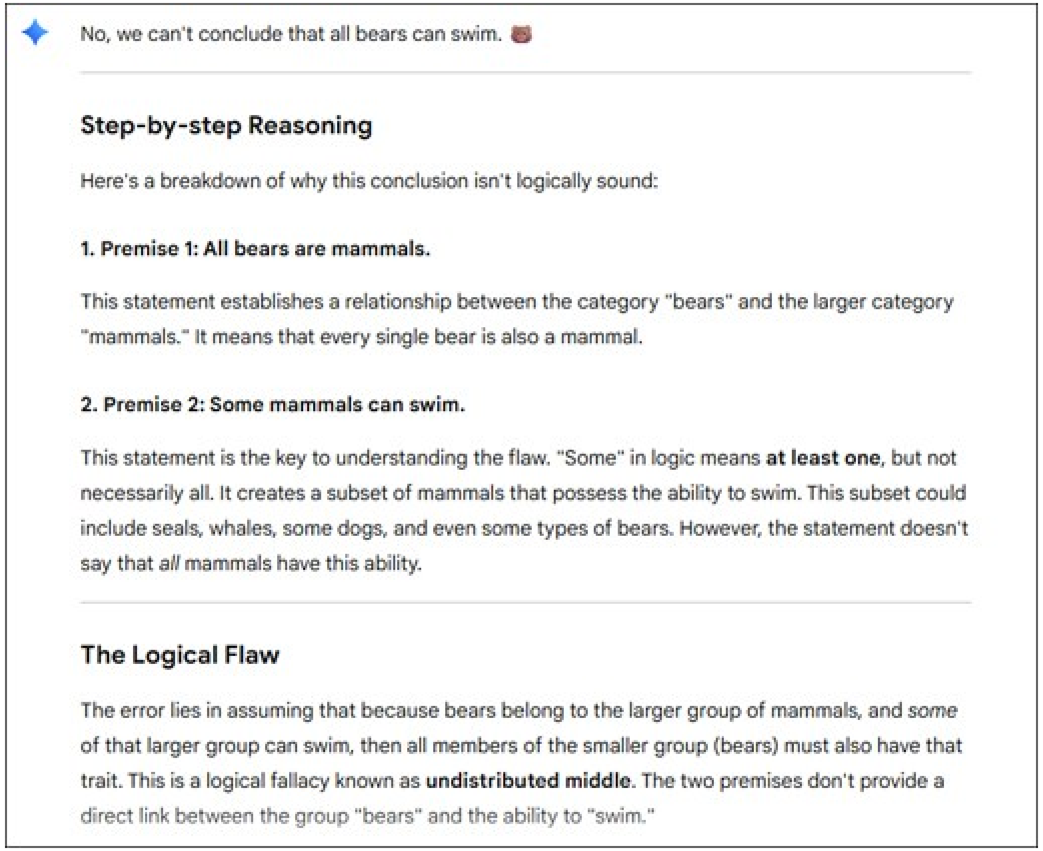
- CoT prompting is the simplest and most popular prompting technique.
- Instead of having the LLM answer immediately, we prompt it to reason step by step.



- This technique usually improves the accuracy because the model reasons through the problem before generating the final output.

Chain of Thought (CoT) Prompt Example

If all bears are mammals, and some mammals can swim, can we conclude that all bears can swim? Provide step-by-step reasoning.



No, we can't conclude that all bears can swim. 🐻

Step-by-step Reasoning

Here's a breakdown of why this conclusion isn't logically sound:

1. Premise 1: All bears are mammals.

This statement establishes a relationship between the category "bears" and the larger category "mammals." It means that every single bear is also a mammal.

2. Premise 2: Some mammals can swim.

This statement is the key to understanding the flaw. "Some" in logic means **at least one**, but not necessarily all. It creates a subset of mammals that possess the ability to swim. This subset could include seals, whales, some dogs, and even some types of bears. However, the statement doesn't say that *all* mammals have this ability.

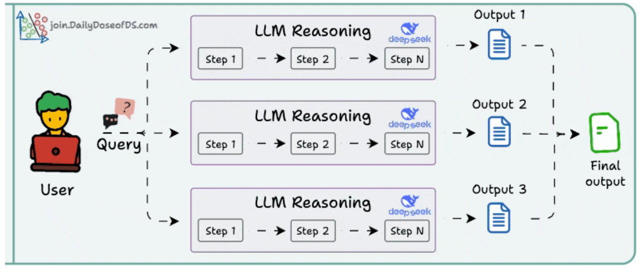
The Logical Flaw

The error lies in assuming that because bears belong to the larger group of mammals, and *some* of that larger group can swim, then all members of the smaller group (bears) must also have that trait. This is a logical fallacy known as **undistributed middle**. The two premises don't provide a direct link between the group "bears" and the ability to "swim."

Carnegie Mellon University

Prompting Strategies: Self-Consistency

- CoT improves reasoning but can be inconsistent. Repeated prompts may yield different answers due to sampling temperature.

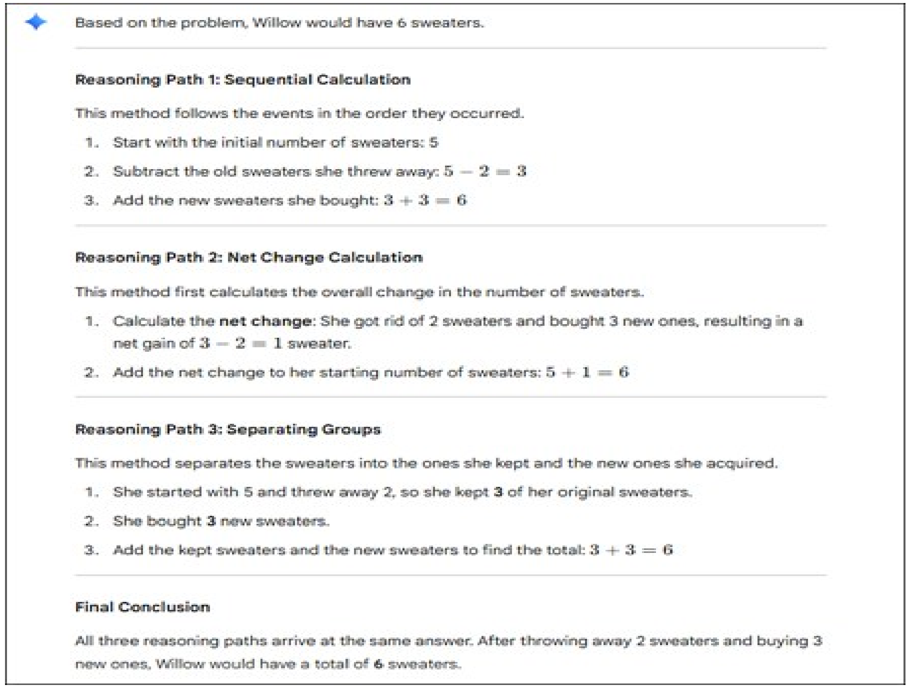


- Self-consistency prompting asks the model to generate multiple independent reasoning paths.
- The most frequent final answer is selected (majority vote).
- The method evaluates agreement among answers, not the quality of reasoning.

Self-Consistency Prompt Example

Given the following problem, generate multiple answers using diverse reasoning paths and aggregate the answers to come to a final conclusion. Use temperature 0.7 to introduce variation.

Willow had 5 sweaters. If she threw away 2 old ones and bought 3 new ones, how many sweaters would she have?



Based on the problem, Willow would have 6 sweaters.

Reasoning Path 1: Sequential Calculation

This method follows the events in the order they occurred.

1. Start with the initial number of sweaters: 5
2. Subtract the old sweaters she threw away: $5 - 2 = 3$
3. Add the new sweaters she bought: $3 + 3 = 6$

Reasoning Path 2: Net Change Calculation

This method first calculates the overall change in the number of sweaters.

1. Calculate the **net change**: She got rid of 2 sweaters and bought 3 new ones, resulting in a net gain of $3 - 2 = 1$ sweater.
2. Add the net change to her starting number of sweaters: $5 + 1 = 6$

Reasoning Path 3: Separating Groups

This method separates the sweaters into the ones she kept and the new ones she acquired.

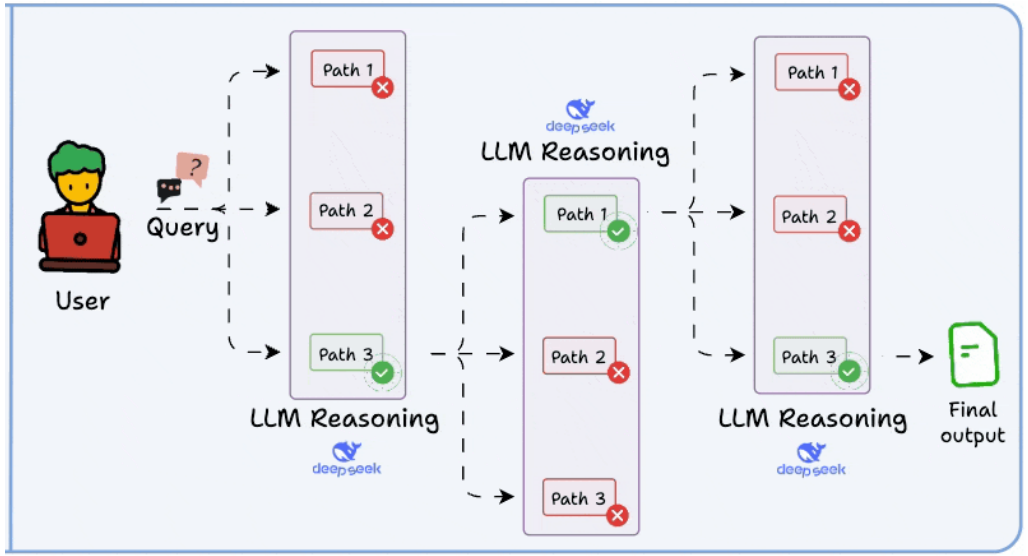
1. She started with 5 and threw away 2, so she kept **3** of her original sweaters.
2. She bought **3** new sweaters.
3. Add the kept sweaters and the new sweaters to find the total: $3 + 3 = 6$

Final Conclusion

All three reasoning paths arrive at the same answer. After throwing away 2 sweaters and buying 3 new ones, Willow would have a total of **6** sweaters.

Prompting Strategies: Tree of Thoughts (ToT)

- Self-consistency varies the final answer, but <u>**Tree of Thoughts varies the reasoning steps themselves**</u>.
- At each step, the LLM explores multiple possible reasoning directions.
- These alternatives form a branching tree of reasoning paths.
- An evaluation process selects the most promising path at each stage.
- The goal is to identify the most coherent path to the solution.
- Tree of Thoughts often outperforms basic CoT.



Tree-of-Thoughts Prompt Example

[TASK TO PERFORM]

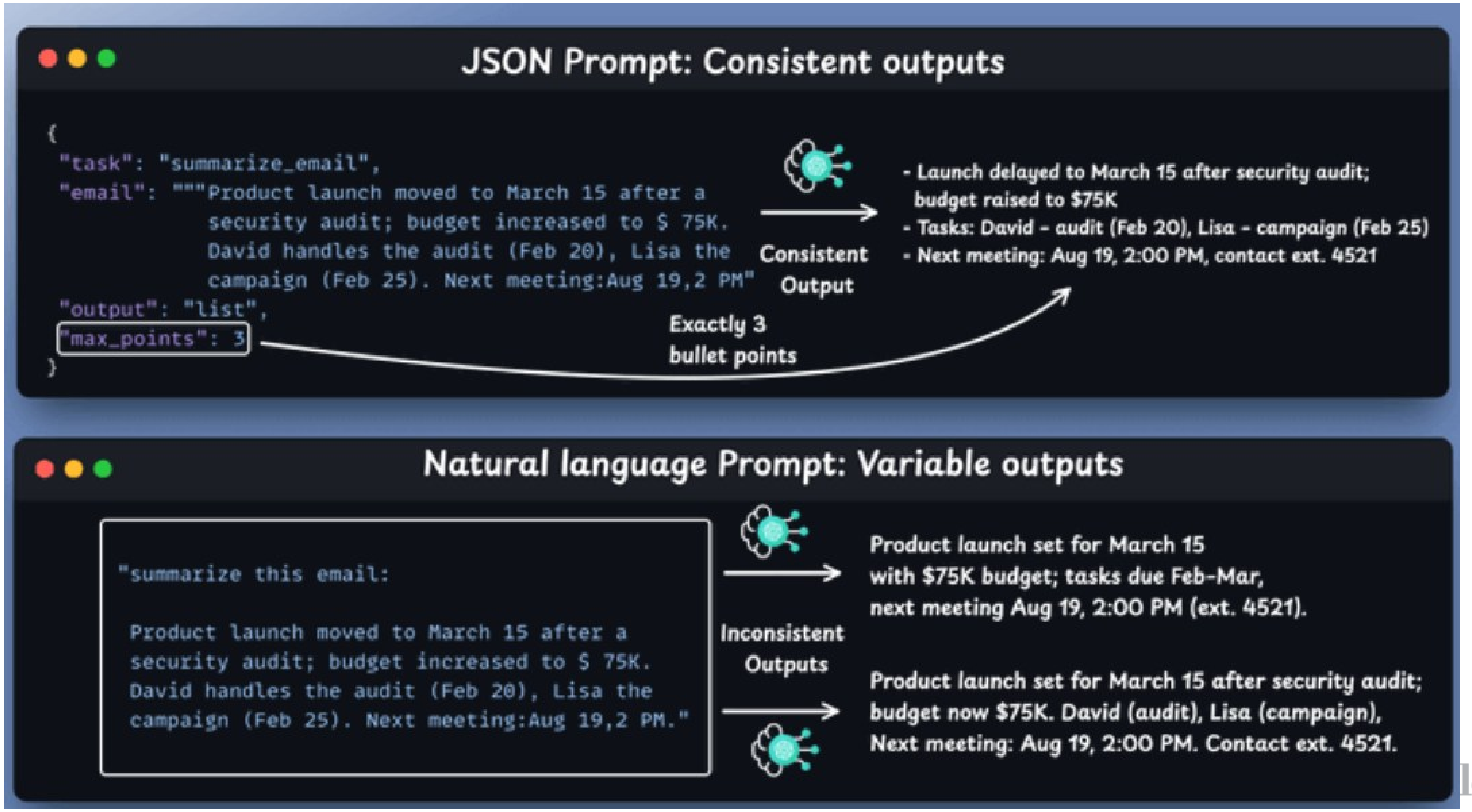
1. Generate at least three different possible approaches.

2. Evaluate each approach.

3. Determine which approach makes the most sense.

4. Proceed to perform the task using the chosen approach.

JSON Prompting vs Text Prompting



JSON Prompt: Consistent outputs

```
{
 "task": "summarize_email",
 "email": """Product launch moved to March 15 after a
          security audit; budget increased to $ 75K.
          David handles the audit (Feb 20), Lisa the
          campaign (Feb 25). Next meeting:Aug 19,2 PM"
 "output": "list",
 "max_points": 3
}
```

- Launch delayed to March 15 after security audit; budget raised to $75K
- Tasks: David - audit (Feb 20), Lisa - campaign (Feb 25)
- Next meeting: Aug 19, 2:00 PM, contact ext. 4521

Natural language Prompt: Variable outputs

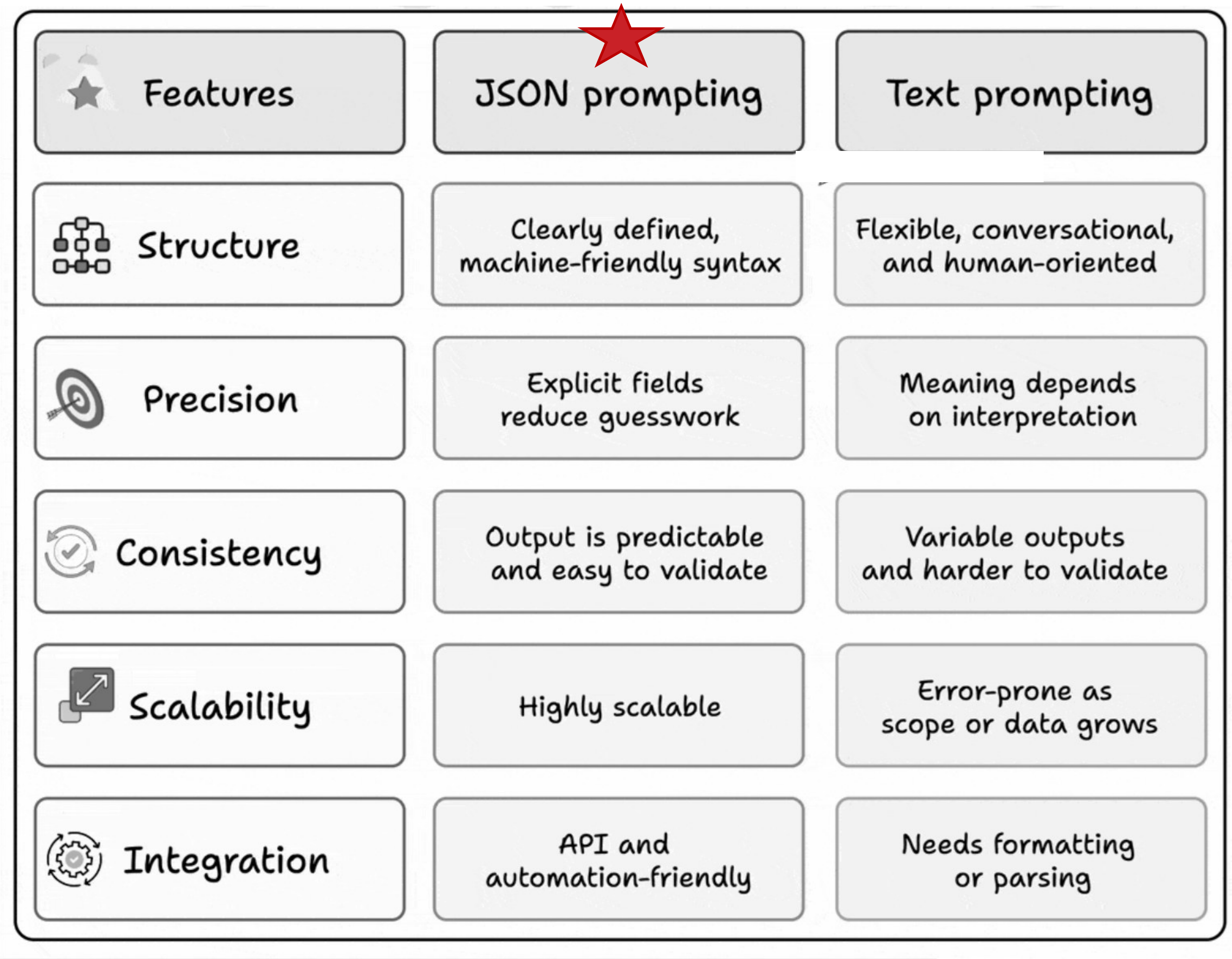
```
"summarize this email:

 Product launch moved to March 15 after a
 security audit; budget increased to $ 75K.
 David handles the audit (Feb 20), Lisa the
 campaign (Feb 25). Next meeting:Aug 19,2 PM."
```

Product launch set for March 15 with $75K budget; tasks due Feb-Mar, next meeting Aug 19, 2:00 PM (ext. 4521).

Product launch set for March 15 after security audit; budget now $75K. David (audit), Lisa (campaign), Next meeting: Aug 19, 2:00 PM. Contact ext. 4521.

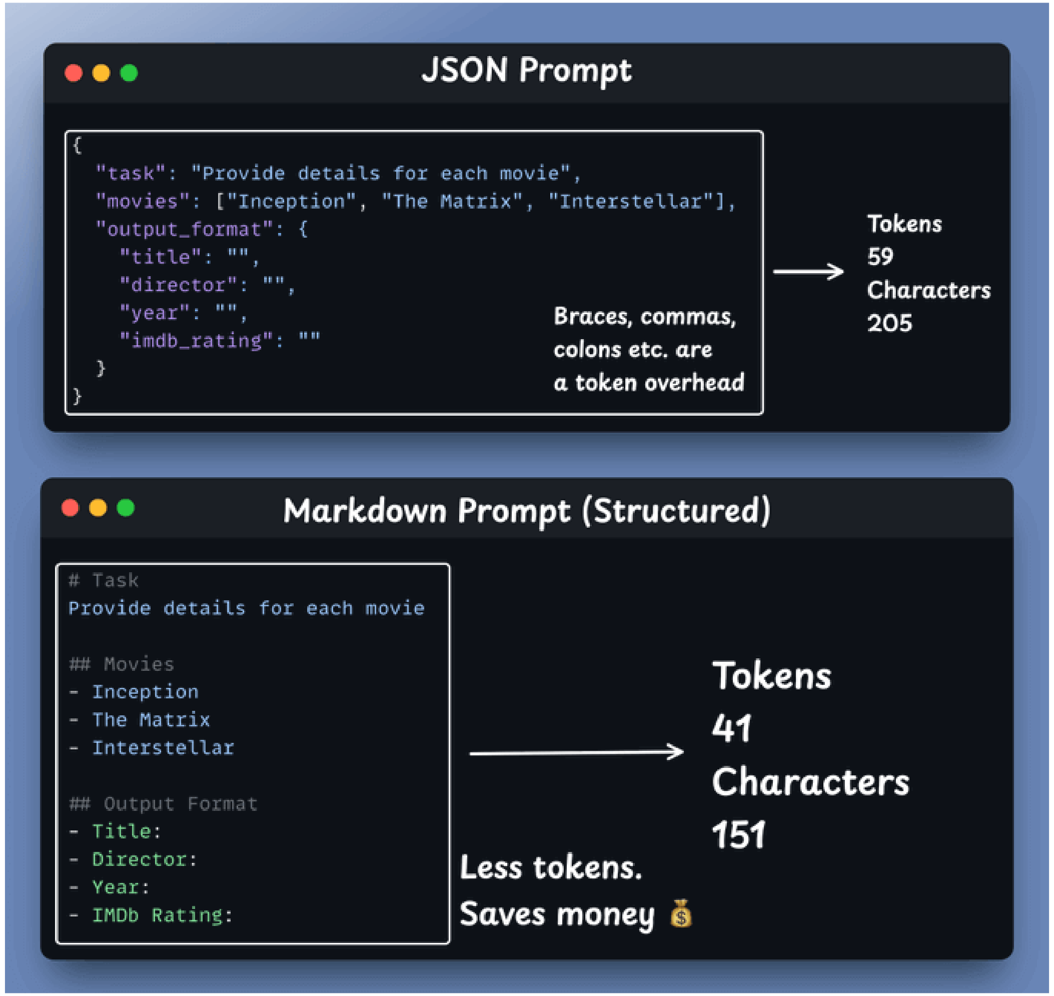
JSON Prompting vs Text Prompting



Features	JSON prompting	Text prompting
Structure	Clearly defined, machine-friendly syntax	Flexible, conversational, and human-oriented
Precision	Explicit fields reduce guesswork	Meaning depends on interpretation
Consistency	Output is predictable and easy to validate	Variable outputs and harder to validate
Scalability	Highly scalable	Error-prone as scope or data grows
Integration	API and automation-friendly	Needs formatting or parsing

JSON Prompting vs Markdown Prompting

Which one is cheaper?



```
{
  "task": "Provide details for each movie",
  "movies": ["Inception", "The Matrix", "Interstellar"],
  "output_format": {
    "title": "",
    "director": "",
    "year": "",
    "imdb_rating": ""
  }
}
```

Braces, commas, colons etc. are a token overhead

Tokens 59 Characters 205

```
# Task
Provide details for each movie

## Movies
- Inception
- The Matrix
- Interstellar

## Output Format
- Title:
- Director:
- Year:
- IMDb Rating:
```

Tokens 41 Characters 151

Less tokens. Saves money 💰

Summary

- Use structured prompting when prompting LLMs.
- Include essential, and ideally optional, prompt components to your prompt.
- Prompt LLMs in JSON or Markdown-like format.
- Prefer Chain-of-Thought prompting for low-cost reasoning, Self-Consistency for problems with multiple plausible solutions, and Tree-of-Thoughts for complex tasks that require exploring alternative reasoning paths.

Readings

- Chain of Thought Prompting: https://www.promptingguide.ai/techniques/cot
- Self-Consistency Prompting: https://www.promptingguide.ai/techniques/consistency
- Tree-of-Thoughts Prompting: https://www.promptingguide.ai/techniques/consistency