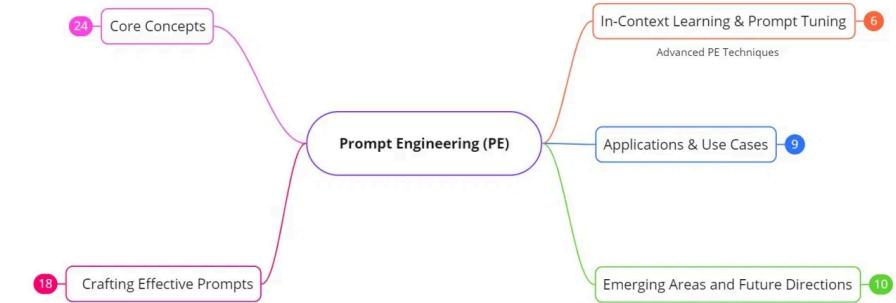




# Prompt and Context Engineering

## Tutorial for Beginners

A Comprehensive Guide to Effective AI Communication



# What is Prompt Engineering?

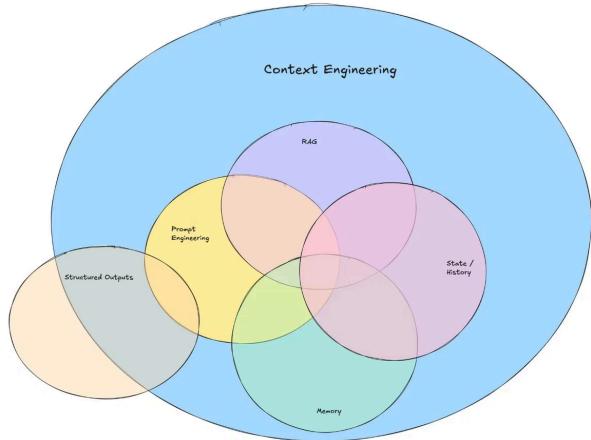
## DEFINITION

Prompt engineering is the art and science of crafting instructions that guide AI language models to produce desired outputs. Think of it as learning to communicate effectively with AI systems to achieve specific goals.

## WHY IT MATTERS

- | No programming skills required to use AI effectively
- | Good prompts dramatically improve AI performance
- | An iterative skill that improves with practice
- | Essential for productivity in many fields

# Prompt vs. Context Engineering



## PROMPT ENGINEERING

### How You Ask

Crafting the instruction you give the model. Focus on wording, structure, roles, constraints, and output schema.

---

## CONTEXT ENGINEERING

### What You Show

Curating the information the model can see. Focus on retrieval, documents, knowledge bases, and tools/APIs.

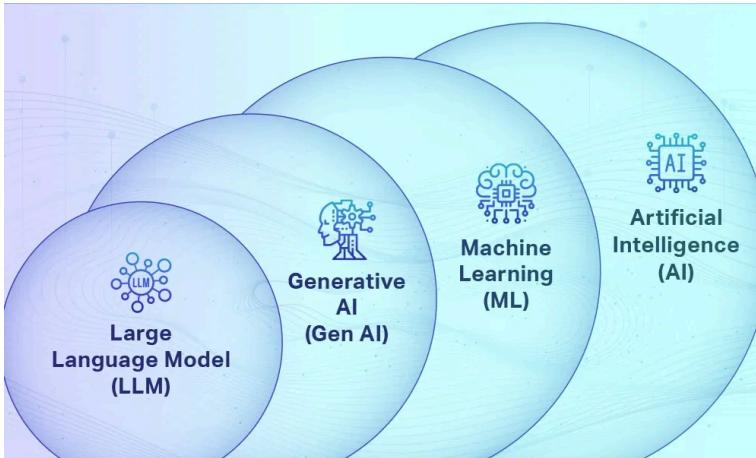
---

## TOGETHER

### The Complete Solution

Combine both for reliable, scalable LLM applications. Prompt guides behavior; context supplies knowledge.

# How LLMs Work



## LLMs AS PREDICTION ENGINES

- ▶ Take text input (your prompt)
- ▶ Predict the next most likely word/token
- ▶ Continue this process to generate complete responses
- ▶ Base predictions on patterns learned from training data

## KEY CONCEPT: AUTOCOMPLETION

LLMs don't "understand" in the human sense—they're **sophisticated autocomplete systems**. Your prompt sets up the context for what should come next.

# Essential Configuration Settings

## TEMPERATURE (0-1)

- Low (0-0.3): Focused, consistent, deterministic
- Medium (0.4-0.7): Balanced creativity and consistency
- High (0.8-1.0): Creative, diverse, unpredictable

Controls the creativity vs. consistency trade-off in responses.

## OUTPUT LENGTH / TOKEN LIMITS

Controls the maximum response length. Higher limits increase computational cost. Set appropriately for your task needs.

## TOP-K AND TOP-P (NUCLEUS SAMPLING)

- Top-K: Limits choices to top K most likely tokens
- Top-P: Limits choices based on cumulative probability

Work together with temperature to control randomness in token selection.

## RECOMMENDED STARTING POINTS

### CONSERVATIVE

Temperature: 0.1  
Top-P: 0.9  
Top-K: 20

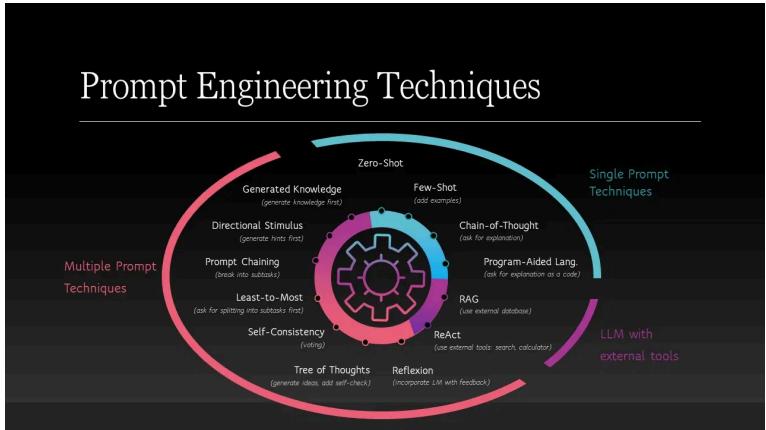
### BALANCED

Temperature: 0.2  
Top-P: 0.95  
Top-K: 30

### CREATIVE

Temperature: 0.9  
Top-P: 0.99  
Top-K: 40

# Fundamental Prompting Techniques



## ZERO-SHOT PROMPTING

Direct request without examples. Use for simple, well-defined tasks where the model has clear domain knowledge.

*Example: "Classify this movie review as positive, negative, or neutral: 'The film was visually stunning but the plot felt rushed.'"*

## ONE-SHOT PROMPTING

Provide a single example to guide the response format and establish the pattern for the model to follow.

*Example: Translate English to French with one example, then ask for another translation.*

## FEW-SHOT PROMPTING

Provide multiple examples (3-5) to establish a clear pattern. Include diverse examples and mix up classes in classification tasks.

*Example: Convert customer feedback to structured JSON with multiple feedback examples.*

# Fundamental Techniques (Part 2)

## SYSTEM PROMPTING

Set overall context and behavior guidelines for the AI model. Define how it should act and what values it should uphold.

### EXAMPLE

*"You are a helpful travel guide. Provide practical, accurate information about destinations."*

## ROLE PROMPTING

Assign a specific character or expertise to the AI. This helps the model adopt the right perspective and knowledge base.

### EXAMPLE

*"Act as an experienced software architect. I need help designing a scalable web application."*

## CONTEXTUAL PROMPTING

Provide specific background information relevant to the task. Give the AI the context it needs to understand your request.

### EXAMPLE

*"Context: You're writing for a tech blog aimed at beginners. Explain APIs in simple language."*

# Chain of Thought Prompting

## WHAT IS COT?

- Encourage step-by-step reasoning for complex problems
- Break down complex tasks into logical sequences
- Improve accuracy and reasoning quality

## WHEN TO USE

- Math problems and calculations
- Logical reasoning tasks
- Complex analysis and decision-making
- Multi-step processes

## HOW TO USE

- Use phrases like "**Let's think step by step**"
- Set temperature to 0 for consistent reasoning
- Extract final answers separately from reasoning

## EXAMPLE

Solve this step by step: If I was 6 when my sister was half my age, how old is my sister when I'm 40? Let me think through this step by step:

# Advanced Techniques

## SELF-CONSISTENCY

### WHAT IT DOES

Generate multiple reasoning paths and select the most common answer.  
Increases reliability by comparing results.

### PROCESS

- ▶ Ask the same question multiple times with different phrasings
- ▶ Compare the answers generated
- ▶ Choose the most frequently occurring result

### BENEFITS

- ▶ Reduces errors from single flawed reasoning paths
- ▶ Increases confidence when paths converge
- ▶ Leverages model's ability to approach problems from different angles

## STEP-BACK PROMPTING

### WHAT IT DOES

Ask a more general question first, then use that context for the specific question. Grounds responses in general knowledge.

### PROCESS

- ▶ Step 1: Ask a broader, foundational question
- ▶ Step 2: Establish key principles or context
- ▶ Step 3: Apply to the specific problem

### BENEFITS

- ▶ Ensures informed and structured recommendations
- ▶ Reduces risk of overlooking critical factors
- ▶ Improves quality of final responses

# ReAct & Tree of Thoughts

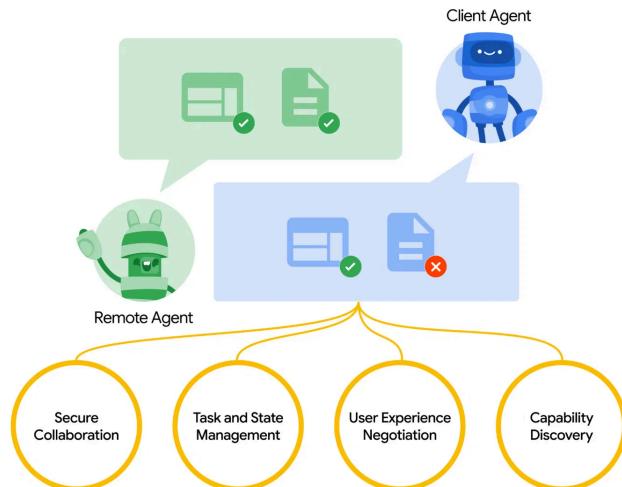
## REACT (REASONING + ACTING)

Combines reasoning with tool use or actions to solve complex problems iteratively.

- Interleaves Thoughts (planning) with Actions (tool calls)
- Processes Observations (results) to refine approach
- Continues cycle until problem is resolved

### USE CASES

- Real-time data retrieval and verification
- Multi-step problem solving
- Dynamic information gathering



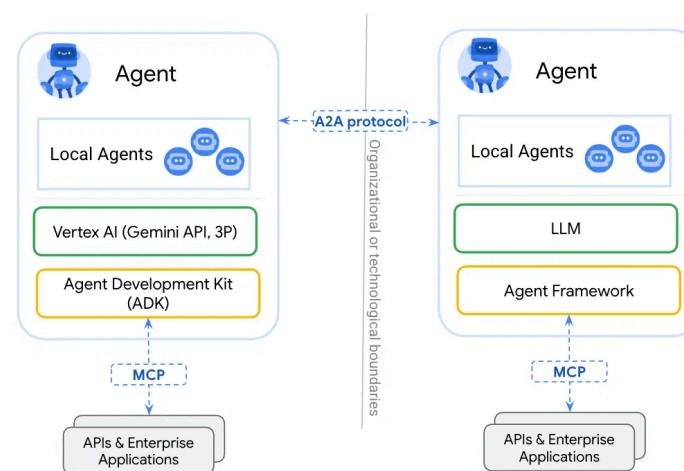
## TREE OF THOUGHTS (TOT)

Explores multiple reasoning branches simultaneously for complex problems.

- Generates multiple solution paths
- Evaluates each branch independently
- Selects the most promising approach

### USE CASES

- Creative problem solving
- Strategic planning and decision-making
- Exploration of alternatives



# Best Practices for Effective Prompts

01

## BE SPECIFIC AND CLEAR

Write clear, unambiguous instructions. Avoid vague language that could be interpreted multiple ways.

04

## ASSIGN A ROLE

Tell the AI what role to adopt. This helps it access the right knowledge and perspective for your task.

02

## PROVIDE CONTEXT & CONSTRAINTS

Give the AI background information and set clear boundaries. Define what you want and what you don't want.

05

## BREAK DOWN COMPLEX TASKS

Divide complex problems into smaller, manageable steps. Use techniques like Chain of Thought for reasoning.

03

## USE EXAMPLES (FEW-SHOT)

Provide 3-5 quality examples to establish the pattern. Include diverse examples that cover edge cases.

06

## TEST & ITERATE

Continuously refine your prompts. Test different approaches and measure results. Improvement comes through experimentation.

# Conclusion & Next Steps

## KEY TAKEAWAYS

- Prompt engineering is a crucial skill for effective AI communication
- Mastering different prompting techniques unlocks the full potential of LLMs
- Context is as important as the prompt itself
- Both prompt and context engineering work together for reliable AI applications

## NEXT STEPS

- Practice with the provided tools and examples
- Explore resources like OpenAI, Google AI Studio, and Anthropic Console
- Keep learning and experimenting with different techniques
- Apply these skills to real-world projects and challenges

Prompt engineering is an iterative skill that improves with practice. Start simple, test your prompts, and continuously refine your

