Introduction to Large Language Models

Spring 2026

Prompt Engineering

(Some slides adapted from Ralph Grishman at NYU,
Yejin Choi at UWashington, N. Tomura at UDepaul, Jurafsky and Martin,
CS224N, CS224, CME295 at Stanford and other resourses on the web)

Prompt Engineering

- **Prompt engineering** is the practice of **designing and refining inputs (prompts)** to guide a Large Language Model (LLM) toward **accurate, reliable, and useful outputs**—*without changing the model's weights*. It is an **inference-time control technique**.
- **Why Prompt Engineering matters**

  LLMs are powerful but **underspecified** by default. A vague prompt yields vague results. Prompt engineering exists to:

  - Clarify **intent** ("what to do")
  - Constrain **behavior** (format, tone, length, safety)
  - Improve **reasoning** (step-by-step, verification)
  - Reduce **hallucinations** (grounding, requirements)
  - Increase **consistency** without retraining

In short: **better prompts → better outcomes**.

Core components of a good prompt

- A strong prompt often includes some combination of:
- **Role** – who the model should act as
  “You are a careful math tutor.”
- **Task** – what to do
  “Solve the problem and explain your reasoning.”
- **Constraints** – how to do it
  Length, format, tone, citations, safety rules
- **Context** – what information to use
  User input, background, retrieved docs (RAG)
- **Examples** – how outputs should look (ICL)
  One-shot / few-shot
- **Process guidance** – how to think (scaffolding)
  Step-by-step, check your work, verify

Prompt Enginering Methods

- Prompt engineering has evolved from a trial-and-error process into a structured discipline. To get the most accurate, reliable, and nuanced outputs from Large Language Models (LLMs), practitioners rely on a core set of proven **methods and patterns**.
- Here is a breakdown of the most common and effective prompt engineering techniques, categorized by their primary function.
  - Foundational Patterns
  - Reasoning and Logic Enhancement
  - Context and Task Management

Foundational Patterns

- **Zero-Shot Prompting:** Instructing the model to perform a task without providing any examples. This relies entirely on the model's pre-trained knowledge and is best used for straightforward, easily understood tasks (e.g., "Summarize this paragraph in two sentences").
- **Few-Shot Prompting:** Providing a small number of example inputs and their corresponding desired outputs within the prompt. This "in-context learning" is highly effective for teaching the model a specific format, tone, or highly specialized classification system that it wouldn't know by default.
- **Role (Persona) Prompting:** Assigning a specific identity, profession, or perspective to the AI (e.g., "Act as a senior database architect explaining SQL to a beginner"). This narrows the model's semantic focus, ensuring it uses the correct industry vocabulary and pitches the explanation at the right level.

Reasoning and Logic Enhancement

- **Chain-of-Thought (CoT):** Asking the model to "think step-by-step" or explicitly outlining the logical steps it needs to take. By forcing the model to generate its reasoning process before producing the final answer, you drastically reduce errors in math, coding, and complex logic puzzles.
- **Tree of Thoughts (ToT):** A more advanced reasoning framework where the model explores multiple different paths to solve a problem, evaluates the viability of each path (often self-correcting), and selects the optimal solution.
- **Self-Consistency:** Generating multiple, independent responses to the exact same reasoning prompt and then having the model select the most consistent or frequently occurring answer as the final output. This acts as a consensus mechanism to filter out anomalies.

Context and Task Management

- **Prompt Chaining:** Breaking a massive, complex task into a sequence of smaller, sequential prompts where the output of step one becomes the input for step two. This prevents the model from experiencing "information overload" and losing track of instructions mid-generation.
- **Generate Knowledge Prompting:** Instructing the model to first recall or generate foundational facts about a topic, and then use those generated facts to answer the actual question. This provides the model with a grounded reference point and reduces hallucinations.
- **Structural Formatting (XML/JSON Structuring):** Using explicit markup to cleanly separate different parts of a prompt. For example, placing background data inside <context> tags, guidelines inside <rules> tags, and asking the model to output the final result in a strict programmatic format like JSON. This is crucial for building reliable software applications around LLMs.

Advanced Optimization

- **Meta-Prompting:** Leveraging the AI to engineer its own prompts. Instead of writing a complex prompt from scratch, you describe your goal to the LLM and ask it to write the most effective, optimized prompt to achieve that specific outcome.
- **Directional Stimulus Prompting:** Providing a specific "hint" or keyword constraint alongside the prompt to steer the model's generation in a highly specific direction (e.g., providing an article to summarize, but explicitly listing the three keywords the summary must revolve around).

General Best Practices

To maximize the effectiveness of any of these patterns, prompts should always prioritize **clarity over politeness** (AI doesn't need "please" or "thank you"), utilize **positive constraints** (tell the model what *to* do, rather than what *not* to do), and be continuously **iterated upon** based on the specific quirks of the model you are using.

Prompting Examples

Zero-Shot Prompting

You simply give the instruction without examples.

Example

Prompt:

```
Explain the difference between evaporation and
boiling in simple terms.
```

Why it works:

The model already knows the concepts and can answer without examples.

Prompting Examples

Few-Shot Prompting

You provide examples to guide the model's behavior.
This is a form of *in-context learning*, but still considered a prompting technique.

Example

Prompt:

```
Translate the following sentences from Turkish to English:

Türkçe: Merhaba, nasılsın?
İngilizce: Hello, how are you?

Türkçe: Bugün hava çok güzel.
İngilizce:
```

Why it works:

The model sees the pattern and continues it.

Prompting Examples

Instruction Prompting

You give a structured command using clear instructions.

Example

Prompt:

```
You are an expert physics instructor.
Explain quantum entanglement to a high school student using simple analogies.
Use a friendly tone and include a real-world metaphor.
```

Why it works:

The model aligns with the specified role, audience, tone,
and format.

Prompting Examples

Chain-of-Thought Prompting

You ask the model to show its reasoning steps
(necessary for complex reasoning).

Example

Prompt:

```
Solve the problem step by step and explain your reasoning:

A train travels 120 km in 2 hours.
What is its average speed?
```

What the model does:

Shows the calculation → then the answer.

Prompting Examples

Self-Consistency *(Wang et al., 2022)*

Sample multiple CoT reasoning paths, then **majority-vote** the final answers. Reduces variance from any single reasoning chain.

```
[Same prompt, temperature > 0, sampled N times]
→ Answer A (×7), Answer B (×2), Answer C (×1)
→ Return A
```

Best for: Tasks with verifiable answers (math, logic) *Cost:* N× inference cost

Prompting Examples

Tree of Thoughts (ToT) *(Yao et al., 2023)*

Generalizes CoT to a **search tree**. The model generates multiple reasoning steps at each node, evaluates them, and explores the most promising branches (BFS or DFS).

```
Problem
├── Approach A
│   ├── Step A1 → [evaluate: promising]
│   └── Step A2 → [evaluate: dead end] ✗
└── Approach B
    └── Step B1 → [evaluate: promising]
        └── Step B2 → Answer ✓
```

Best for: Planning, puzzles, tasks requiring backtracking *Cost:* Significantly more inference calls

Prompting Examples

Persona or Role Prompting

You tell the model to behave like a specific expert,
character, or identity.

Example

Prompt:

```
You are a senior data scientist.
Explain what PCA is, including intuition and a mathematical example.
```

Prompting Examples

Template / Structured Prompting

Useful for tasks requiring consistent output formatting.

Example

Prompt:

```
Summarize the text below using this format:
- Key Points:
- Important Terms:
- One-Sentence Summary:

Text: "Machine learning is a subset of AI..."
```

Prompting Examples

ReAct (Reason + Act) *(Yao et al., 2022)*

Interleaves **reasoning traces** with **tool actions** in a loop:

```
Thought: I need to find the population of Tokyo.
Action: search("Tokyo population 2024")
Observation: Tokyo has ~13.9 million in the city proper.
Thought: Now I can answer.
Answer: Tokyo's population is approximately 13.9 million.
```

Foundation for most modern agent frameworks (LangChain, LlamaIndex)

Prompting Examples

Deliberate Prompting

Ask the model to think before answering.

Example

Prompt:

```
Think carefully about the question before answering.
Consider at least 3 possible explanations.
Then choose the most logical one.

Question: Why do objects float in water?
```

Key Components of a Good Prompt

To write an effective prompt, consider including:

☑ 1. Role

Tell the model who it is.

```
You are a legal expert.
```

☑ 2. Task

What you want.

```
Summarize this document.
```

☑ 3. Context

Background information.

```
This document is a contract about intellectual property.
```

☑ 4. Constraints

How you want the answer.

```
Use bullet points and keep it under 150 words.
```

☑ 5. Examples (optional)

If you need consistent formatting.

```
Example output:
- Summary:
- Risks:
```

A High-Quality Prompt Example

Prompt:

```
You are an expert educator.
Explain the concept of entropy to a university freshman.

Requirements:
- Use simple, intuitive language.
- No equations.
- Use one analogy and one real engineering example.
- Keep the explanation under 150 words.

 Example style:
 "Think of it like..."
```

Prompting Templates

Writing & Summarization

```
Rewrite the following text with:
- professional tone
- clear logic
- improved readability
- no change in meaning

TEXT:
```

```
Summarize the text in:
- 5 bullet points
- 1-sentence takeaway
```

Prompting Templates

Data Analysis

```
You are a data analysis assistant.
Explain the dataset below, including:
- Variables
- Insights
- Potential issues
- Recommended next steps
```

Prompting Templates

Engineering/Technical Explanation

```
Explain the concept of [TOPIC] to:
- a beginner
- an engineering student
- an expert

Provide 3 different levels of explanation.
```