Prompt Engineering: Crafting Effective AI Interactions

Dr Eniafe Ayetiran

July 2025

Outline

What is Prompt Engineering?

- **Definition**: The process of designing and refining input prompts to guide AI models (e.g., LLMs) to produce desired outputs.

What is Prompt Engineering?

- **Definition**: The process of designing and refining input prompts to guide AI models (e.g., LLMs) to produce desired outputs.
- **Goal**: Optimize model performance without changing the model itself.

What is Prompt Engineering?

- **Definition**: The process of designing and refining input prompts to guide AI models (e.g., LLMs) to produce desired outputs.
- **Goal**: Optimize model performance without changing the model itself.
- **Why Important?**: Enables precise, efficient, and creative interactions with AI.

Why Prompt Engineering Matters

- **Efficiency**: Well-crafted prompts reduce trial-and-error.

Why Prompt Engineering Matters

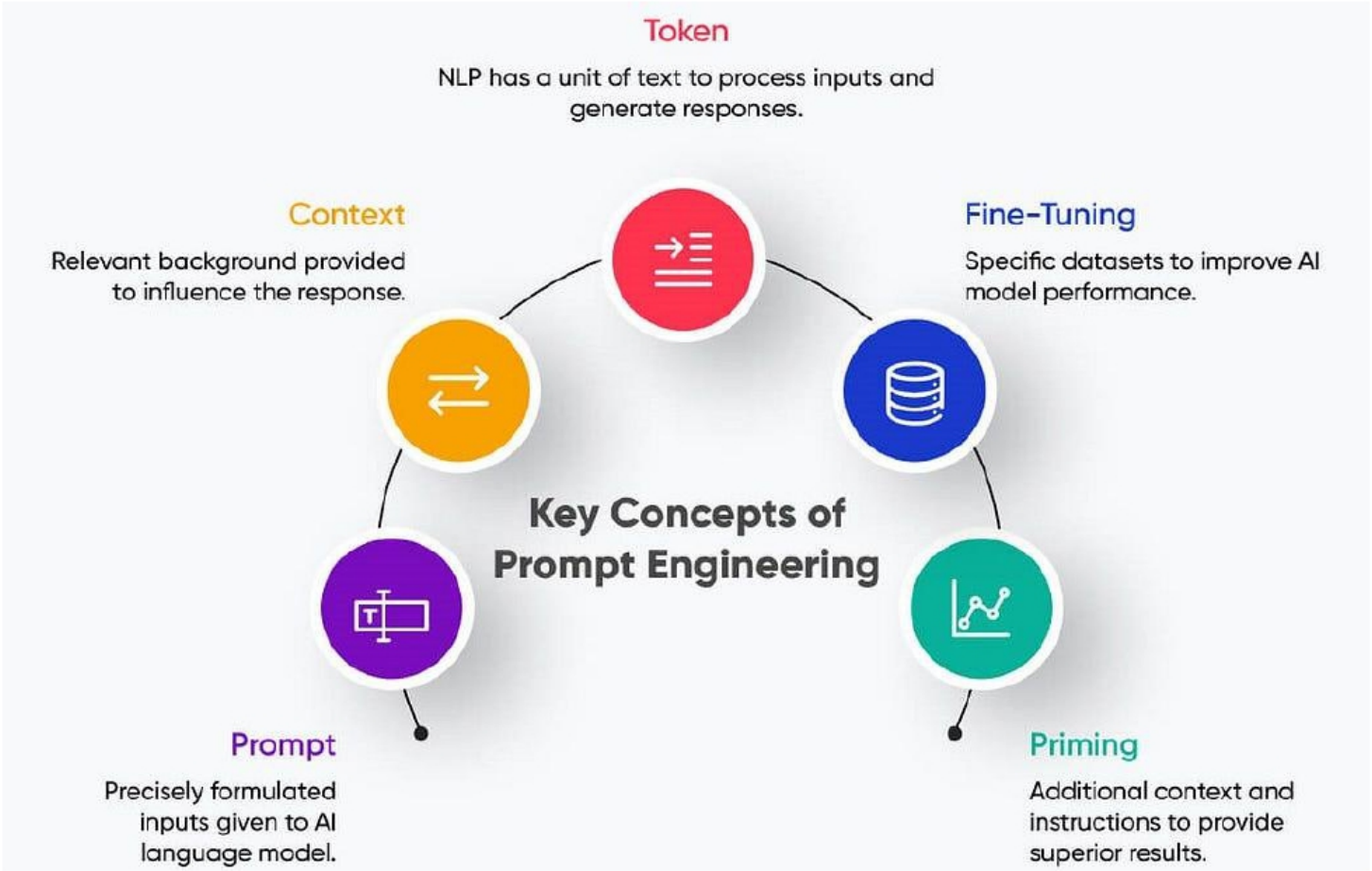
- **Efficiency**: Well-crafted prompts reduce trial-and-error.
- **Accessibility**: Allows non-experts to leverage complex AI models.

Why Prompt Engineering Matters

- **Efficiency**: Well-crafted prompts reduce trial-and-error.
- **Accessibility**: Allows non-experts to leverage complex AI models.
- **Applications**: Content creation, problem-solving, data analysis, and more.

Why Prompt Engineering Matters

- **Efficiency**: Well-crafted prompts reduce trial-and-error.
- **Accessibility**: Allows non-experts to leverage complex AI models.
- **Applications**: Content creation, problem-solving, data analysis, and more.
- **Connection to AI**: Core to interacting with LLMs like ChatGPT, Grok.


Token
NLP has a unit of text to process inputs and generate responses.
Context
Relevant background provided to influence the response.
Fine-Tuning
Specific datasets to improve AI model performance.
Key Concepts of Prompt Engineering
Prompt
Precisely formulated inputs given to AI language model.
Priming
Additional context and instructions to provide superior results.

PRACTICAL EXAMPLES OF PROMPT ENGINEERING

- “Write an introduction for a blog post about the benefits of remote work.”
- 'As a customer support agent, how would you respond to a customer who asks for a refund due to a defective product?'
- 'Create a lesson plan for a high school biology class on the topic of cell division.'
- 'Write a short story about a robot learning to understand human emotions.'
- 'Analyze the financial performance of Company X for the past fiscal year.'

Key Components of a Prompt

- **Instruction**: Clear task or question (e.g., “Summarize this text”).

Key Components of a Prompt

- **Instruction**: Clear task or question (e.g., "Summarize this text").
- **Context**: Background information to guide the model.

Key Components of a Prompt

- **Instruction**: Clear task or question (e.g., “Summarize this text”).
- **Context**: Background information to guide the model.
- **Examples**: Sample inputs/outputs to clarify expectations.

Key Components of a Prompt

- **Instruction**: Clear task or question (e.g., "Summarize this text").
- **Context**: Background information to guide the model.
- **Examples**: Sample inputs/outputs to clarify expectations.
- **Constraints**: Specific requirements (e.g., "Use 50 words").

Types of Prompts

- **Zero-Shot**: Direct task without examples (e.g., “Translate this to French”).

Types of Prompts

- **Zero-Shot**: Direct task without examples (e.g., “Translate this to French”).
- **Few-Shot**: Provide a few examples to guide output (e.g., “Here are two summaries, write a third”).

Types of Prompts

- **Zero-Shot**: Direct task without examples (e.g., “Translate this to French”).
- **Few-Shot**: Provide a few examples to guide output (e.g., “Here are two summaries, write a third”).
- **Chain-of-Thought (CoT)**: Encourage step-by-step reasoning (e.g., “Explain your reasoning”).

Types of Prompts

- **Zero-Shot**: Direct task without examples (e.g., “Translate this to French”).
- **Few-Shot**: Provide a few examples to guide output (e.g., “Here are two summaries, write a third”).
- **Chain-of-Thought (CoT)**: Encourage step-by-step reasoning (e.g., “Explain your reasoning”).
- **Role-Based**: Assign a role (e.g., “Act as a teacher”).

Prompt Engineering Workflow

- **Step 1**: Define the task clearly.

Prompt Engineering Workflow

- **Step 1**: Define the task clearly.
- **Step 2**: Craft an initial prompt.

Prompt Engineering Workflow

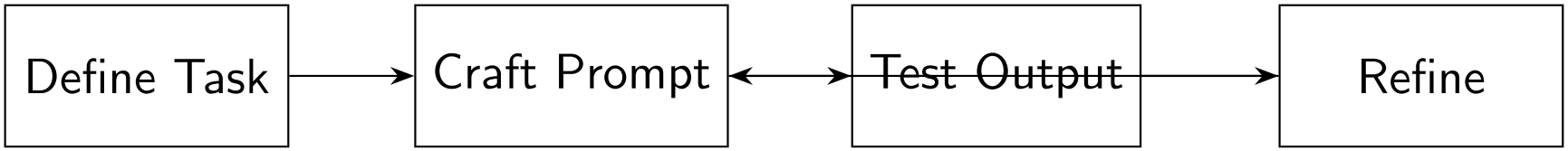
- **Step 1**: Define the task clearly.
- **Step 2**: Craft an initial prompt.
- **Step 3**: Test and evaluate output.

Prompt Engineering Workflow

- **Step 1**: Define the task clearly.
- **Step 2**: Craft an initial prompt.
- **Step 3**: Test and evaluate output.
- **Step 4**: Refine prompt based on results.

Prompt Engineering Workflow

- **Step 1**: Define the task clearly.
- **Step 2**: Craft an initial prompt.
- **Step 3**: Test and evaluate output.
- **Step 4**: Refine prompt based on results.
- **Visualization**:



Crafting Effective Prompts

- **Clarity**: Use precise, unambiguous language.

Crafting Effective Prompts

- **Clarity**: Use precise, unambiguous language.
- **Specificity**: Define output format (e.g., list, paragraph).

Crafting Effective Prompts

- **Clarity**: Use precise, unambiguous language.
- **Specificity**: Define output format (e.g., list, paragraph).
- **Context**: Provide relevant background (e.g., dataset description).

Crafting Effective Prompts

- **Clarity**: Use precise, unambiguous language.
- **Specificity**: Define output format (e.g., list, paragraph).
- **Context**: Provide relevant background (e.g., dataset description).
- **Iterative Refinement**: Adjust based on model responses.

Chain-of-Thought Prompting

- **Concept**: Ask the model to reason step-by-step.

Chain-of-Thought Prompting

- **Concept**: Ask the model to reason step-by-step.
- **Example**:
  - Prompt: "Solve $2x + 3 = 7$. Show your steps."
  - Output: "Subtract 3: $2x = 4$. Divide by 2: $x = 2$."

Chain-of-Thought Prompting

- **Concept**: Ask the model to reason step-by-step.
- **Example**:
  - Prompt: "Solve $2x + 3 = 7$. Show your steps."
  - Output: "Subtract 3: $2x = 4$. Divide by 2: $x = 2$."
- **Benefit**: Improves accuracy for complex tasks.

Few-Shot Prompting

- **Concept**: Provide examples to guide the model.

Few-Shot Prompting

- **Concept**: Provide examples to guide the model.
- **Example**:
  - Prompt: "Classify: 'I love this!' (Positive). 'It's okay' (Neutral). 'Terrible service' (?)."
  - Output: Negative.

Few-Shot Prompting

- **Concept**: Provide examples to guide the model.
- **Example**:
  - Prompt: "Classify: 'I love this!' (Positive). 'It's okay' (Neutral). 'Terrible service' (?)."
  - Output: Negative.
- **Benefit**: Helps model understand task without explicit training.

Role-Based Prompting

- **Concept**: Assign a persona to the model.

Role-Based Prompting

- **Concept**: Assign a persona to the model.
- **Example**:
  - Prompt: "Act as a history professor. Explain the French Revolution."
  - Output: Detailed, academic explanation.

Role-Based Prompting

- **Concept**: Assign a persona to the model.
- **Example**:
  - Prompt: "Act as a history professor. Explain the French Revolution."
  - Output: Detailed, academic explanation.
- **Benefit**: Tailors tone and style to specific needs.

Example: Text Summarization

- **Prompt**: "Summarize this article in 50 words: [article text]."

Example: Text Summarization

- **Prompt**: "Summarize this article in 50 words: [article text]."
- **Output**: Concise summary capturing key points.

Example: Text Summarization

- **Prompt**: "Summarize this article in 50 words: [article text]."
- **Output**: Concise summary capturing key points.
- **Key Elements**: Clear instruction, word limit constraint.

Example: Text Summarization

- **Prompt**: "Summarize this article in 50 words: [article text]."
- **Output**: Concise summary capturing key points.
- **Key Elements**: Clear instruction, word limit constraint.
- **Refinement**: If too vague, add "Focus on main arguments."

Example: Data Analysis

- **Prompt**: "Given sales data [1, 2, 3, 4, 5], calculate the mean and variance. Show steps."

Example: Data Analysis

- **Prompt**: "Given sales data [1, 2, 3, 4, 5], calculate the mean and variance. Show steps."
- **Output**: Step-by-step calculation (mean = 3, variance = 2).

Example: Data Analysis

- **Prompt**: "Given sales data [1, 2, 3, 4, 5], calculate the mean and variance. Show steps."
- **Output**: Step-by-step calculation (mean = 3, variance = 2).
- **Key Elements**: CoT, specific dataset, step-by-step request.

Applications in Industry

- **Content Creation**: Blogs, marketing copy.

Applications in Industry

- **Content Creation**: Blogs, marketing copy.
- **Customer Support**: Chatbots with tailored responses.

Applications in Industry

- **Content Creation**: Blogs, marketing copy.
- **Customer Support**: Chatbots with tailored responses.
- **Education**: Generating study materials, tutoring.

Applications in Industry

- **Content Creation**: Blogs, marketing copy.
- **Customer Support**: Chatbots with tailored responses.
- **Education**: Generating study materials, tutoring.
- **Programming**: Code generation, debugging.

Applications in Research

- **Data Analysis**: Automating statistical tasks (e.g., hypothesis testing).

Applications in Research

- **Data Analysis**: Automating statistical tasks (e.g., hypothesis testing).
- **NLP Research**: Testing LLM capabilities.

Applications in Research

- **Data Analysis**: Automating statistical tasks (e.g., hypothesis testing).
- **NLP Research**: Testing LLM capabilities.
- **Synthetic Data**: Generating datasets for ML training.

Applications in Research

- **Data Analysis**: Automating statistical tasks (e.g., hypothesis testing).
- **NLP Research**: Testing LLM capabilities.
- **Synthetic Data**: Generating datasets for ML training.
- **Scientific Writing**: Drafting papers, summarizing literature.

Challenges in Prompt Engineering

- **Ambiguity**: Vague prompts lead to poor outputs.

Challenges in Prompt Engineering

- **Ambiguity**: Vague prompts lead to poor outputs.
- **Model Limitations**: LLMs may misinterpret complex prompts.

Challenges in Prompt Engineering

- **Ambiguity**: Vague prompts lead to poor outputs.
- **Model Limitations**: LLMs may misinterpret complex prompts.
- **Bias**: Prompts can amplify biases in model training data.

Challenges in Prompt Engineering

- **Ambiguity**: Vague prompts lead to poor outputs.
- **Model Limitations**: LLMs may misinterpret complex prompts.
- **Bias**: Prompts can amplify biases in model training data.
- **Trial-and-Error**: Finding the right prompt requires iteration.

Best Practices

- **Be Specific**: Clearly state the task and output format.

Best Practices

- **Be Specific**: Clearly state the task and output format.
- **Provide Context**: Include relevant background information.

Best Practices

- **Be Specific**: Clearly state the task and output format.
- **Provide Context**: Include relevant background information.
- **Use Examples**: Few-shot prompts for clarity.

Best Practices

- **Be Specific**: Clearly state the task and output format.
- **Provide Context**: Include relevant background information.
- **Use Examples**: Few-shot prompts for clarity.
- **Iterate**: Test and refine prompts based on results.

Key Takeaways

- Prompt engineering is the art of crafting inputs to optimize AI outputs.

Key Takeaways

- Prompt engineering is the art of crafting inputs to optimize AI outputs.
- Techniques like CoT, few-shot, and role-based prompting enhance performance.

Key Takeaways

- Prompt engineering is the art of crafting inputs to optimize AI outputs.
- Techniques like CoT, few-shot, and role-based prompting enhance performance.
- Applications span industries, from content creation to research.

Key Takeaways

- Prompt engineering is the art of crafting inputs to optimize AI outputs.
- Techniques like CoT, few-shot, and role-based prompting enhance performance.
- Applications span industries, from content creation to research.
- **Future**: Prompt engineering will evolve with advancing LLMs.

References

- Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33.

References

- Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33.
- Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*.