Module 2

Chapter:01 Introduction to Prompt Engineering

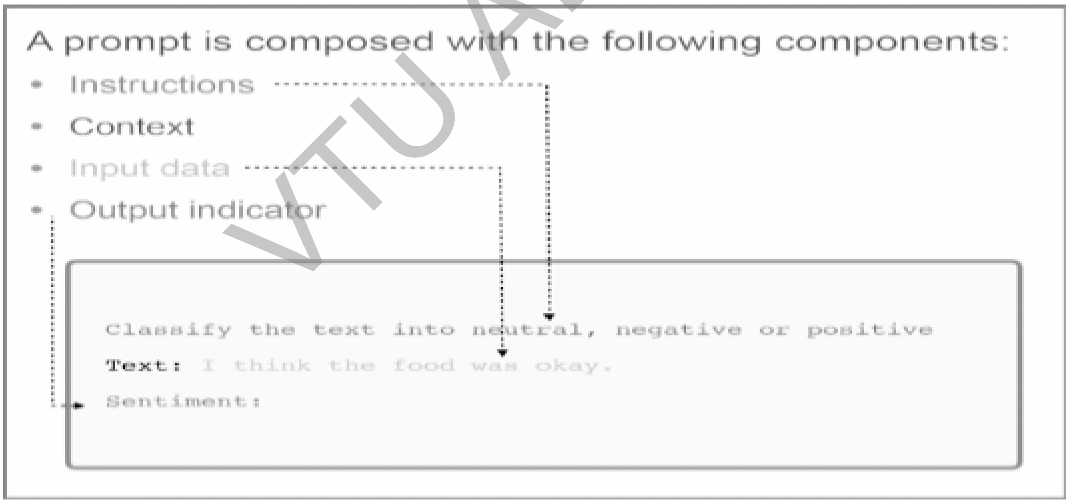
Chapter:02 Prompt Engineering Techniques for ChatGPT

1. Define Prompt Engineering and explain its components with an example

Definition:
Prompt engineering is the art and science of designing input instructions (prompts) that guide Large Language Models (LLMs) such as ChatGPT to produce accurate, creative, and context-appropriate outputs. A *prompt* serves as a roadmap that establishes context, specifies the task, and provides cues or constraints to influence how the AI responds.

Components:

1. **Instruction**– Describes what the model must do (e.g., "Summarize," "Translate," "Classify").
2. **Context**– Gives relevant information or scenario details to anchor the model's reasoning.
3. **Input Data** /– States the specific query or content on which the model must act.
4. **Output Format**– Specifies tone, structure, or style expected in the answer.



A prompt is composed with the following components:

- Instructions
- Context
- Input data
- Output indicator

```
Classify the text into neutral, negative or positive
Text: I think the food was okay.
Sentiment:
```

Fig. 2 Components of Prompt

Example:
Prompt → *"Summarize the following paragraph in three bullet points, highlighting the main causes of climate change."*

Here, the **instruction** ("Summarize"), **context** ("causes of climate change"), and **format** ("three bullet points") together ensure a precise, structured response.

2. What Makes Prompt Engineering Vital?

Prompt engineering is crucial because it enables us to achieve the best possible results from Large Language Models (LLMs) like ChatGPT. It enables users to control how the AI behaves, making it useful for various creative and practical purposes.
The main reasons why prompt engineering is vital are:

- **Goes beyond built-in limits:**
  Prompts help us use the full power of LLMs for things they were not directly trained for — such as writing stories, poems, or even composing music.
- **Improves accuracy and control:**
  By designing prompts carefully, we can guide the AI to give correct, relevant, and factual answers. This is especially important in areas like journalism, education, or research.
- **Boosts creativity and originality:**
  Well-written prompts can spark the AI's imagination to generate new ideas, creative solutions, and unique content such as artworks or inventions.
- **Makes AI accessible to everyone:**
  Prompt engineering allows people without technical or programming knowledge to use AI easily. With simple and clear prompts, anyone can use AI for learning, writing, designing, or problem solving.

3. How can prompt engineering make advanced AI systems more accessible to non-technical users?

Prompt engineering *democratizes AI* by enabling anyone—without coding or data science expertise—to interact with LLMs using plain language.

- **User-friendly interaction:** Natural-language prompts replace complex programming syntax.
- **Guided behavior:** Clear prompts steer models to perform specific tasks, reducing trial and error.
- **Customization:** Templates and prompt libraries enable reuse for writing, research, or design.
- **Empowerment:** Writers, teachers, or business users can harness AI for creative or analytical work simply by phrasing effective questions.
  Thus, prompt engineering bridges the gap between human intent and AI capability, expanding AI's reach to a wider audience.

4. List and explain the types of prompts

There are three main types of prompts used in prompt engineering. Each type serves a specific purpose, but all aim to guide the Large Language Model (LLM) to generate accurate, relevant, and well-structured outputs.

1. Natural-Language Prompts

- These are prompts written in simple, everyday language, just like how humans communicate naturally.

- To make AI understand instructions clearly and respond in a natural, human-like manner.
- **Example:**
  "Explain Newton's laws in simple words."

This prompt uses plain English, so the AI understands that it must provide a simple explanation suitable for beginners.

- Natural-language prompts are best for writing, teaching, summarizing, and general question–answer tasks because they don't require any coding or special format.

2. **System Prompts**

- These are predefined or hidden instructions given to the model to control its tone, role, or behavior throughout a session.
- They act like "rules" that guide how the model should communicate.
- To maintain consistency and control in how the AI responds, especially in professional or formal settings.
- **Example:**
  System prompt: "You are a polite customer-service assistant. Always respond kindly and offer helpful information."
- System prompts are widely used in chatbots, customer service bots, and digital assistants, ensuring the model's responses stay polite, factual, and professional.

3. **Conditional Prompts**

- Conditional prompts include specific conditions or logic statements that tell the AI what to do based on a situation or input.
- They often use "if–then" structures.
- To help the model adapt its response depending on the given context or data.
- **Example:**
  "If the input mentions temperature, suggest suitable clothing."
  This means the AI should only provide clothing advice when the topic of temperature is mentioned.
- Conditional prompts are important for decision-making systems, recommendation tools, and context-aware chatbots, where responses must change based on conditions.

5. **Explain the working principles of prompt engineering/ How Does Prompt Engineering Work?**

- Prompt engineering works through a **cyclic and experimental process**. There is no single rule that works for every task because every Large Language Model (LLM) and use case is different.
- Therefore, prompt engineers follow an **iterative (step-by-step testing and refining)** approach to design prompts that guide the model toward the desired results.

1. **Understand the Task**

- Before writing a prompt, it is important to **clearly define the goal** of the task.
- You must decide what type of output you expect — for example, a summary, a poem, a translation, or a code snippet.

- Understanding the task helps the engineer frame the prompt that directs the model correctly.
- **Example:** If the goal is to generate a news headline, the prompt should specify *"Write a short, catchy headline for the given news paragraph."*

2. **Design the Prompt**

- The next step is to create a clear, specific, and concise prompt.
- Ambiguous or confusing instructions can mislead the model.
- Simple and direct wording improves accuracy.
- **Tip:** Avoid technical jargon unless necessary; use straightforward instructions.
- **Example:** Instead of "Discuss environmental degradation," use "Write three sentences describing the causes of air pollution."

3. **Provide Context and Examples**

- Adding context or sample outputs helps the model understand the structure, tone, or detail level required.
- The more relevant information you include, the more accurate the response will be.
- Providing examples (as in *one-shot* or *few-shot* prompting) teaches the model what kind of answer is expected.
- **Example:** "Example: Input – '2+2'; Output – '4'. Now solve: 5+3 = ?" This helps the model follow a pattern.

4. **Test and Refine**

- Prompt engineering is an **experimental process**.
- You should try multiple versions of a prompt, compare their outputs, and see which gives the best result.
- This "trial and error" helps improve accuracy, reduce errors, and make the output more natural.
- **Textbook Note:** "There is no one-size-fits-all method. Experimenting with various prompts is the best way to discover what functions."

5. **Evaluate Performance**

- After testing, evaluate how well the prompt performs in terms of:
  - **Accuracy** – Is the information correct?
  - **Creativity** – Is the output original or insightful?
  - **Coherence** – Is the response logical and well-structured?
- Based on evaluation, further refinements can be made to improve results.

6. How does prompt engineering improve communication between humans and LLMs?

- Communication is the foundation of all human interaction, and Prompt Engineering has the power to revolutionize the way humans communicate with AI systems like Large Language Models (LLMs). By designing effective prompts, users can make AI express ideas more clearly, creatively, and ethically.
- The major roles of prompt engineering in communication are explained below:

1. **Improving Specificity and Clarity**

- Prompts act like **guiding lights** that help LLMs produce clear, meaningful, and easy-to-understand responses.
- By providing proper **context, goal, and tone**, users can ensure that the message delivered by the model is **accurate and audience-friendly**.
- This avoids confusion or misunderstanding between the AI and the user.
- **Example:** Instead of saying "Write about pollution," a clear prompt like "Write a short paragraph explaining the causes of air pollution for school students" gives a more focused result.

2. Increasing Creativity and Engagement

- Well-crafted prompts can **unlock the creative potential** of AI.
- They can encourage the LLM to produce **imaginative stories, poems, debates, or ideas** that inspire thought and discussion.
- Prompts can also make communication more interactive and enjoyable.
- **Example:** A prompt like "Write a conversation between a teacher and a robot about the future of learning" leads to engaging, creative dialogue.

3. Adapting Communication to Different Contexts

- Every situation requires a different **tone and style** of communication.
- Prompts allow the model to **adjust its language** depending on the purpose — for example, formal writing for a report, persuasive writing for an argument, or casual tone for chatting.
- This helps AI communicate naturally across various professional and personal contexts.
- **Example:** "Write a formal email to your manager explaining project progress" vs. "Write a friendly message to your friend about your new project."

4. Overcoming Linguistic Barriers

- Prompt engineering helps LLMs **translate languages accurately and naturally**, promoting **cross-cultural understanding**.
- By including cultural hints or intended meanings in prompts, users can guide AI to interpret words and expressions correctly.
- This makes global communication smoother and more effective.
- **Example:** A prompt like "Translate this paragraph into Hindi while keeping a polite and respectful tone" helps preserve cultural nuance.

5. Addressing Ethical Issues

- Although prompt engineering is powerful, it must be used **responsibly and ethically**.
- Poorly designed prompts can include **biases or stereotypes**, leading to unfair or harmful outputs.
- Prompts should be created carefully to **avoid discrimination, misinformation, or manipulation**.
- Ethical prompt design ensures that communication remains fair, safe, and trustworthy.

7. Explain the Advantages and Applications of Prompt Engineering

Advantages of Prompt Engineering

1. **Improves Accuracy and Control:** Carefully written prompts help LLMs give precise and factual answers, reducing errors and confusion.
2. **Enhances Creativity:** Prompts inspire AI to generate new ideas, stories, and artistic or innovative content.
3. **Saves Time and Effort:** Natural-language prompts allow users to complete complex tasks quickly without coding.
4. **Accessible to Non-Technical Users:** Anyone can use AI easily through simple language, making technology more user-friendly.
5. **Personalization and Ethical Use:** Prompts can be designed to produce responses in a desired tone or style while avoiding bias or misuse.

Applications of Prompt Engineering

1. **Education:** Used to create summaries, quizzes, and explanations for students and teachers.
2. **Content Creation:** Writers and marketers use prompts to generate blogs, captions, and creative material.
3. **Customer Service:** System prompts control chatbot behavior to provide polite and helpful replies.
4. **Programming and Research:** Prompts help generate or debug code, summarize data, and produce technical explanations.
5. **Translation and Decision Support:** Used for accurate multilingual communication and generating business or research insights.

8. Explain how Zero-shot, One-shot, Few-shot, and Self-consistency prompting techniques can improve LLM performance in real-world tasks

Prompt engineering utilizes various prompting strategies to enhance the performance of Large Language Models (LLMs) across diverse real-world applications, including translation, problem-solving, summarization, coding, and reasoning. Each method provides a different level of guidance to the model through examples or reasoning paths.

1. Zero-Shot Prompting

- The model is given only the instructions, without any examples. It depends entirely on its pre-trained general knowledge to generate a response.
- Helps test how well the model can understand and perform new or unseen tasks without prior examples.
- Example:
  "Write a short poem about nature."
  The model uses its existing knowledge of poetry and nature to create an answer.
- Suitable for creative writing, summarization, and question answering, where specific examples are not required.

2. One-Shot Prompting

- The model is given one example before performing the task. This single example helps the model understand the expected style, tone, or structure.
- Helps improve the accuracy and relevance of responses when a small hint or pattern is provided.
- Example:
  "Example: Translate 'Bonjour' → 'Hello'. Now translate 'Au revoir'."
  After seeing one translation, the model easily follows the same pattern.
- Useful for translation, classification, and short text generation tasks, where one example is enough to guide the output format.

3. Few-Shot Prompting

- The model is provided with a few (2–5) examples before being asked to perform the main task.
- It helps the model learn the pattern, logic, or structure of responses by studying examples.
- Example:
  Giving multiple math problem–solution pairs and then asking the model to solve a new one helps it follow the same reasoning.
- Ideal for complex reasoning tasks, question answering, sentiment analysis, and data formatting, where pattern recognition is needed.

4. **Self-Consistency Prompting**

- The model is asked to generate multiple reasoning paths for the same question and then select or average the most consistent and logical answer.
- To improve reasoning accuracy and reduce random or incorrect outputs.
- Example:
  In solving a logic problem, the model may generate several solutions and then choose the one that appears most frequently or is logically sound.
- Helpful in analytical writing, problem-solving, decision-making, and reasoning-based tasks, ensuring the final output is consistent and reliable.

Summary Table

Technique	**Examples Given**	**Purpose / Benefit**	**Application Area**
Zero-Shot	None	Uses general knowledge	Creative writing, summarization
One-Shot	1 example	Learns pattern or tone	Translation, classification
Few-Shot	Few examples	Improves logical accuracy	Problem-solving, reasoning
Self-Consistency	Multiple reasoning paths	Selects most logical output	Decision-making, analysis

9. Explain the Evolution of Prompt Engineering

Evolution of Prompt Engineering

- 1966 – ELIZA: Keyword-based chatbot mimicking conversation, showing language can guide AI behavior.
- 1970s – SHRDLU: Reasoned about objects in a virtual world via user commands, enabling task-directed interaction.
- 1990s – Jabberwacky & ALICE: Generated humorous, open-ended responses, exploring creative prompt use.
- 2000s – Template-based systems (Madalyn, Galatea): Structured user input to generate narratives, establishing organized prompting methods.
- 2017 – Transformer revolution: Enabled LLMs like GPT-3, making prompt engineering key for precise, human-like outputs.
- 2020s – Modern LLM optimization: Focus on task-specific prompt adaptation and fine-grained control over outputs.

10. Describe the categories of prompts in the Prompt Gallery and explain how they support different NLP tasks.

The Prompt Gallery is a curated collection of prompts that helps users guide NLP models like ChatGPT to perform specific tasks effectively. Each category targets a particular NLP application:

1. Text Generation Prompts

Explanation: These prompts instruct the model to create original text in a specific style or format. They can generate stories, essays, product descriptions, or poetry. Users can define the tone, length, and style of the output.

Example: “Write a motivational paragraph encouraging students to study consistently.”

Support for NLP Tasks: Facilitates creative writing, automated content creation, and storytelling applications.

2. Text Classification – Sentiment Analysis Prompts

Explanation: These prompts enable the model to determine the overall sentiment (positive, negative, neutral) in a text. They help analyze opinions and emotional tone.

Example: “Analyze the sentiment of this review: ‘The customer service was very helpful and friendly.’”

Support for NLP Tasks: Useful in market research, social media monitoring, and customer feedback analysis.

3. Text Classification – Sentiment Rating Prompts

Explanation: Similar to sentiment analysis, but instead of categorical labels, the model assigns a numerical score to quantify sentiment.

Example: “Rate this movie review on a scale of 1 to 5 stars: ‘The storyline was compelling, but the acting was average.’”

Support for NLP Tasks: Provides more nuanced insights into audience opinions and preferences.

4. Information Extraction – Entity Extraction Prompts

Explanation: Guide the model to identify and extract key entities such as people, organizations, dates, and locations from text.

Example: “Extract all names, companies, and dates mentioned in this press release.”

Support for NLP Tasks: Enables named-entity recognition, document indexing, and structured data extraction for analysis.

5. **Information Extraction – Relation Extraction Prompts**

Explanation: These prompts help the model detect relationships between entities, like partnerships, family connections, or cause-effect relations.

Example: “Identify the relationships between the people and organizations mentioned in this article.”

Support for NLP Tasks: Useful for building knowledge graphs, automated reasoning, and advanced information retrieval.

6. **Question Answering – Closed-Domain QA Prompts**

Explanation: These restrict the model’s answers to a specific topic or domain, ensuring precision and relevance.

Example: “Answer customer questions about the warranty period of our laptops.”

Support for NLP Tasks: Ideal for customer support systems, help desks, and domain-specific knowledge retrieval.

7. **Question Answering – Open-Domain QA Prompts**

Explanation: Open-domain prompts allow the model to answer questions across a wide range of topics using general knowledge.

Example: “Who discovered penicillin and in which year?”

Support for NLP Tasks: Useful for virtual assistants, educational tools, and general information retrieval.

8. **Text Summarization Prompts**

Explanation: Direct the model to condense lengthy text while retaining essential ideas and information.

Example: “Summarize this 3-page research article into a 6-sentence paragraph.”

Support for NLP Tasks: Helps in document summarization, news aggregation, and quick content review.

9. **Keyword Extraction Prompts**

Explanation: Instruct the model to identify significant words or phrases that capture the main topics of a text.

Example: “Extract the main keywords from this blog post about renewable energy.”

Support for NLP Tasks: Supports search engine optimization (SEO), tagging, and indexing of content.

10. **Text Editing Prompts**

Explanation: Guide the model to improve clarity, grammar, coherence, and style of the text.

Example: “Proofread this paragraph and improve sentence structure and vocabulary.”

Support for NLP Tasks: Automates text refinement, proofreading, and style enhancement.

11. **Text Translation Prompts**

Explanation: Direct the model to translate text accurately between languages while maintaining context and meaning.

Example: “Translate the following English paragraph into Spanish, keeping the tone formal.”

Support for NLP Tasks: Supports multilingual communication, content localization, and cross-lingual applications.

12. Text Transliteration Prompts

Explanation: Help convert text from one script to another while preserving pronunciation, often used for names, technical terms, or cultural phrases.

Example: “Transliterate ‘Ramesh’ from English to Hindi script.”

Support for NLP Tasks: Useful in cross-lingual communication, name recognition, and cultural content adaptation.