1. Write simple prompts to extract summaries, structural answers and lists from an LLM.

1. Aim

To understand the basics of prompt writing and use simple prompts to generate summaries, structured answers, and lists using a Large Language Model (LLM).

2. Objectives

- To study the concept of prompt engineering
- To write simple prompts for summarization
- To generate structured responses using prompts
- To create list-based outputs through prompt instructions

3. Problem Statement

Large Language Models may produce unclear or unstructured responses when prompts are vague. This exercise focuses on learning how to frame clear and precise prompts to obtain concise summaries, structured explanations, and organized lists.

4. Theory / Background

Prompt engineering refers to the method of designing inputs (prompts) that guide a language model to generate meaningful and relevant responses. A well-defined prompt specifies the task, the required output format, and the level of detail. Simple prompts focus on one requirement at a time and reduce ambiguity, resulting in improved output quality.

5. Procedure

Step 1: Summarization Prompt

Sample Prompt

```
Summarize the following text in 4–5 sentences.
```

Explanation The prompt clearly specifies the task and restricts the length of the response, enabling concise summarization.

Expected Output

- Condensed version of the given text
- Limited to 4–5 sentences

Step 2: Structured Answer Prompt

Sample Prompt

Explain Machine Learning using the following structure:

1. Definition

2. Types

3. Applications

Explanation The prompt enforces a structured format, helping organize information logically.

Expected Output

- Explanation divided into defined sections
- Clear and organized presentation

Step 3: List-Based Prompt

Sample Prompt

List 5 applications of Artificial Intelligence.

Explanation This prompt instructs the model to produce information in a list format with a fixed number of items.

Expected Output

- Five-item list
- Bullet-point or numbered format

Step 4: Prompt with Constraints

Sample Prompt

```
Explain cloud computing in simple terms for a beginner in 5 lines.
```

Explanation: The prompt controls complexity and length, making the output easy to understand.

Expected Output

- Simple explanation
- Output restricted to five lines

6. Result

After completing this exercise, we will be able to:

- Frame clear and simple prompts for different tasks
- Control output format, structure, and length
- Understand the importance of prompt clarity in obtaining accurate responses

2. Practice prompt tuning to control response tone, length and specificity

1. Aim

To practice prompt tuning techniques that control the tone, length, and level of detail of responses generated by a Large Language Model (LLM).

2. Objectives

- To understand the concept of prompt tuning
- To control the length of LLM responses
- To modify response tone using prompt instructions
- To adjust specificity and detail in generated outputs

3. Problem Statement

Basic prompts often result in responses that are either too long, too vague, or inappropriate for the intended audience. This exercise focuses on refining prompts by adding constraints to control tone, length, and specificity of the output.

4. Theory / Background

Prompt tuning is a technique used to refine prompts by adding explicit constraints such as word limits, tone specification, audience definition, and level of detail. These constraints help reduce ambiguity and enable the model to generate responses that are more accurate, concise, and suitable for specific use cases.

5. Procedure

Step 1: Controlling Response Length

Sample Prompts

1. Explain Artificial Intelligence in not more than 50 words.
2. Describe cloud computing in exactly 5 lines.
3. Summarize Machine Learning in 3 bullet points.

Expected Output

- Responses limited to the specified length
- Concise and focused content

Step 2: Controlling Response Tone

Sample Prompts

1. Explain data science in a formal academic tone.
2. Explain Artificial Intelligence in a friendly and conversational tone.
3. Describe cloud computing in a professional business tone.

Expected Output

- Responses reflecting the specified tone
- Appropriate language style based on instruction

Step 3: Controlling Specificity Based on Audience

Sample Prompts

1. Explain Machine Learning in simple terms for a school student.
2. Explain Artificial Intelligence for a non-technical audience.
3. Explain cloud computing for computer science students.

Expected Output

- Audience-appropriate explanations
- Adjusted vocabulary and complexity

Step 4: Controlling Level of Detail

Sample Prompts

1. Explain data science briefly and mention only its core components.
2. Explain Artificial Intelligence with detailed explanations and examples.
3. Give a high-level overview of Machine Learning without technical details.

Expected Output

- Responses matching the requested detail level
- Relevant and focused information

6. Result

After completing this exercise, we are able to tune prompts effectively to control response tone, length, and specificity, resulting in clear, relevant, and context-appropriate outputs from a Large Language Model.

3. Create Zero-Shot and Few-Shot Prompts for Tasks like Classification and Q&A

1. Aim

To understand and apply zero-shot and few-shot prompting techniques for performing classification and question-answering tasks using a Large Language Model (LLM).

2. Objectives

- To understand the concept of zero-shot prompting
- To understand the concept of few-shot prompting
- To create classification prompts without examples
- To create classification and Q&A prompts with examples

3. Problem Statement

In many real-world scenarios, it is not always possible to provide training data or fine-tune a model. Instead, users rely on prompts to guide the model's behavior. This exercise focuses on learning how to perform tasks such as classification and question answering using zero-shot and few-shot prompting techniques.

4. Theory / Background

Zero-shot prompting refers to instructing a language model to perform a task without providing any examples. The model relies solely on the task description. Few-shot prompting involves providing a small number of examples within the prompt to demonstrate the expected behavior. These techniques are widely used for tasks such as text classification, sentiment analysis, and question answering, where examples help improve accuracy and consistency.

5. Procedure

Step 1: Zero-Shot Prompting for Classification

Sample Prompts

1. Classify the sentiment of the following text as Positive, Negative, or Neutral.
2. Categorize the following email as Spam or Not Spam.

3. Identify whether the following statement is a Fact or an Opinion.

Expected Output

- Single correct class label
- Clear and concise classification

Step 2: Zero-Shot Prompting for Question Answering

Sample Prompts

1. Answer the following question in one sentence.
2. Answer the following question clearly and concisely.
3. Provide a direct answer to the following question.

Expected Output

- Direct and relevant answers
- No unnecessary explanation

Step 3: Few-Shot Prompting for Classification

Sample Prompts

1. Text: "I love this product." → Sentiment: Positive
Text: "This is the worst service ever." → Sentiment: Negative
Text: "The product is average." → Sentiment: ?

2. Email: "Win a free prize now!" → Spam
Email: "Meeting scheduled for tomorrow." → Not Spam
Email: "Congratulations, you have won!" → ?

3. Statement: "Water boils at 100°C." → Fact
Statement: "This movie is amazing." → Opinion
Statement: "The weather is nice today." → ?

Expected Output

- Correct classification based on given examples
- Consistent labeling pattern

Step 4: Few-Shot Prompting for Question Answering

Sample Prompts

```
1. Q: What is AI?
   A: AI is the simulation of human intelligence in machines.
   Q: What is Machine Learning?
   A:

2. Q: Who invented the telephone?
   A: Alexander Graham Bell.
   Q: Who invented the computer?
   A:

3. Q: What is cloud computing?
  A: It is the delivery of computing services over the internet.
  Q: What is data science?
  A:
```

Expected Output

- Answers following the pattern of given examples
- Clear and concise responses

6. Result

After completing this exercise, students are able to create zero-shot and few-shot prompts for classification and question-answering tasks. The exercise demonstrates how providing clear task instructions or examples improves the accuracy and consistency of responses generated by a Large Language Model.

4. Implement Chain-of-Thought Prompting to Solve Math or Logic Problems

1. Aim

To understand and apply chain-of-thought prompting to enable a Large Language Model (LLM) to solve mathematical and logical problems through step-by-step reasoning.

2. Objectives

- To understand the concept of chain-of-thought prompting
- To guide the model to show intermediate reasoning steps
- To solve mathematical problems using step-by-step prompts
- To solve logical reasoning problems using structured thinking

3. Problem Statement

LLMs may produce incorrect answers for math or logic problems when asked to provide only final results. Without explicit reasoning instructions, intermediate steps are often skipped. This exercise focuses on using chain-of-thought prompting to encourage step-by-step reasoning, improving accuracy and clarity of solutions.

4. Theory / Background

Chain-of-thought prompting is a technique in which the model is instructed to break down a problem into smaller reasoning steps before arriving at a final answer. By explicitly asking the model to “think step by step” or “show all reasoning,” complex problems become easier to solve and verify. This approach is particularly effective for arithmetic calculations, word problems, and logical reasoning tasks.

5. Procedure

Step 1: Chain-of-Thought Prompting for Arithmetic Problems

Sample Prompts

```
1. Solve the following problem step by step:

  If a book costs ₹120 and you buy 3 books, what is the total cost?
```

2. Solve step by step:

   What is the result of (45 × 6) − 30?

3. Show all steps and calculate:

   A train travels 60 km in 1 hour. How far will it travel in 3.5 hours?

Expected Output

- Clear step-by-step calculations
- Correct final numerical answer

Step 2: Chain-of-Thought Prompting for Word Problems

Sample Prompts

1. Read the problem carefully and solve step by step:

   Ramesh has 20 apples. He gives 7 apples to his friend. How many apples remain?

2. Solve the following word problem with proper reasoning:

   A shop gives a discount of ₹50 on an item priced at ₹450. What is the final price?

3. Explain each step clearly while solving:

   If the perimeter of a square is 40 cm, find the length of one side.

Expected Output

- Logical breakdown of the problem
- Clearly explained intermediate steps

Step 3: Chain-of-Thought Prompting for Logical Reasoning

Sample Prompts

1. Solve the following logic problem step by step:

   All cats are animals. Some animals are pets. Can we conclude that some cats are pets?

2. Think step by step and answer:

   If today is Monday, what day will it be after 10 days?

3. Explain your reasoning before giving the answer:

   A is older than B. B is older than C. Who is the youngest?

Expected Output

- Step-by-step logical reasoning
- Clearly stated final conclusion

Step 4: Explicit Reasoning Instruction

Sample Prompts

1. Solve the following problem and explain your reasoning before giving the final answer.

2. Break the problem into steps and solve it logically.

3. Show your full chain of thought while solving this math problem.

Expected Output

- Detailed reasoning process
- Transparent and verifiable solution

6. Result

After completing this exercise, students are able to apply chain-of-thought prompting to guide a Large Language Model through step-by-step reasoning. This approach improves accuracy and clarity when solving mathematical calculations and logical reasoning problems.

5. Use Function Calling in an LLM to Simulate API Tools like Calculator or Search

1. Aim

To understand how function calling can be used in a Large Language Model (LLM) to simulate external tools such as calculators and search utilities.

2. Objectives

- To understand the concept of function calling in LLMs
- To simulate calculator operations using function-based prompts
- To simulate search-like responses using structured function calls
- To observe how LLMs decide when to invoke a function

3. Problem Statement

LLMs are primarily text-based and do not inherently perform real-time calculations or searches. To extend their capabilities, function calling allows LLMs to interact with external tools or APIs. This exercise focuses on designing prompts that simulate such tool usage through structured function calls.

4. Theory / Background

Function calling is a technique where an LLM is guided to return structured outputs (such as JSON-like responses) that represent calls to external functions. These functions may perform tasks like calculations, database queries, or web searches. By defining the function's purpose and parameters clearly in the prompt, the model can decide when and how to invoke the function, enabling tool-assisted reasoning and automation.

5. Procedure

Step 1: Simulating a Calculator Function

Sample Prompts

1. Use a calculator function to add 45 and 78.

2. Call the calculator tool to multiply 12 and 9.

3. Use a calculator function to find the square of 15.

Expected Output

- Structured response indicating a calculator function call
- Correct numerical result

Step 2: Simulating a Search Function

Sample Prompts

1. Use a search function to find the capital of Australia.

2. Call the search tool to retrieve information about Python programming.

3. Use a search function to get the definition of cloud computing.

Expected Output

- Structured response indicating a search function call
- Relevant and concise information

Step 3: Function Calling with Parameters

Sample Prompts

1. Call the calculator function with parameters: number1 = 25, number2 = 5, operation = division.

2. Use the calculator tool with inputs 100 and 40 to perform subtraction.

3. Call the search function with the query parameter "Machine Learning applications".

Expected Output

- Function name with clearly defined parameters
- Correct output based on provided inputs

Step 4: Decision-Based Function Invocation

Sample Prompts

1. If the query requires calculation, use the calculator function; otherwise, answer normally.

2. Decide whether to call a calculator or search function to answer the following query.

3. Choose the appropriate function to solve the given task and return the result.

Expected Output

- Appropriate function selection
- Correct and structured function response

6. Result

After completing this exercise, students are able to design prompts that simulate function calling in LLMs. The exercise demonstrates how structured prompts enable the model to interact with external tools such as calculators and search utilities, improving task accuracy and automation capability.

6. Create Multi-Turn Conversation Prompts Preserving Chat Context

1. Aim

To understand and create multi-turn conversation prompts that preserve context across interactions with a Large Language Model (LLM).

2. Objectives

- To understand the concept of multi-turn conversations
- To maintain conversational context across multiple prompts
- To design follow-up prompts that depend on previous responses
- To observe how context influences LLM responses

3. Problem Statement

Single-turn prompts do not retain context from previous interactions, which limits their usefulness in conversations, interviews, and support systems. Multi-turn conversation prompting allows the model to remember and utilize earlier context, enabling coherent and meaningful interactions. This exercise focuses on designing prompts that preserve and build upon chat context.

4. Theory / Background

Multi-turn conversation prompting involves a sequence of related prompts where each prompt depends on the context established by previous interactions. By maintaining conversation history, the LLM can generate responses that are consistent, relevant, and context-aware. This approach is essential for chatbots, virtual assistants, customer support systems, and interactive tutoring applications.

5. Procedure

Step 1: Establishing Initial Context

Sample Prompts

1. You are a career counselor. Explain what data science is.

2. Act as a travel assistant and suggest a 3-day trip plan to Goa.

3. You are a technical support agent. Help me troubleshoot a slow laptop.

Expected Output

- Context-setting response
- Role-specific explanation or guidance

Step 2: Context-Based Follow-Up Questions

Sample Prompts

1. Based on your previous explanation, list the key skills required.

2. Modify the trip plan to include budget-friendly options.

3. Suggest the next troubleshooting step.

Expected Output

- Responses that reference previous context
- Continuity in tone and topic

Step 3: Refining the Conversation

Sample Prompts

1. Explain the most important skill in more detail.

2. Update the plan by removing one location.

3. Provide a simpler solution suitable for a beginner.

Expected Output

- Refined responses aligned with prior discussion
- Consistent conversational flow

Step 4: Context Preservation Across Turns

Sample Prompts

1. Summarize everything we discussed so far.

2. Based on our conversation, give a final recommendation.

3. Answer using the same role and context as before.

Expected Output

- Summary or conclusion using accumulated context
- Coherent and complete response

6. Result

After completing this exercise, students are able to create multi-turn conversation prompts that preserve context across interactions. The exercise demonstrates how maintaining conversational history improves coherence, relevance, and usefulness of responses generated by a Large Language Model.

7. Design Persona-Based Prompts for Scenarios like Interviews or Tech Support

1. Aim

To design persona-based prompts that enable a Large Language Model (LLM) to respond according to a specific role or character in real-world scenarios such as interviews and technical support.

2. Objectives

- To understand the concept of persona-based prompting
- To define roles and responsibilities within a prompt
- To create prompts for interview scenarios
- To design prompts for customer and technical support use cases

3.Problem Statement

LLMs generate generic responses when no role or context is specified. In real-world applications such as interviews, customer service, and technical support, responses must align with a specific persona. This exercise focuses on designing prompts that define a clear role, enabling the model to respond appropriately and consistently.

4. Theory / Background

Persona-based prompting is a technique where a specific role, behavior, or identity is assigned to the LLM within the prompt. By defining the persona—such as an interviewer, technical support agent, or customer care executive—the model adapts its tone, language, and response style. This technique is widely used in conversational agents, interview simulations, and support systems to create realistic and task-oriented interactions.

5. Procedure

Step 1: Persona-Based Interview Prompts

Sample Prompts

1. You are a technical interviewer. Ask me three interview questions on Python.

2. Act as an HR interviewer and conduct a short interview for a software developer role.

3. You are a senior data scientist. Evaluate my answer to the following question.

Expected Output

- Interview-style questions or feedback
- Professional and role-specific tone

Step 2: Persona-Based Technical Support Prompts

Sample Prompts

1. You are a technical support agent. Help me fix a Wi-Fi connectivity issue.

2. Act as a customer support executive and guide me to reset my account password.

3. You are an IT helpdesk assistant. Diagnose a laptop that is not turning on.

Expected Output

- Step-by-step assistance
- Clear and helpful instructions

Step 3: Persona-Based Customer Interaction Prompts

Sample Prompts

1. You are a customer care representative. Respond politely to a product complaint.

2. Act as a support chatbot and explain the refund policy.

3. You are a service agent. Handle an angry customer professionally.

Expected Output

- Polite and empathetic responses
- Scenario-appropriate language

Step 4: Maintaining Persona Consistency

Sample Prompts

1. Continue responding as the same interviewer throughout the conversation.

2. Maintain the technical support role and provide the next troubleshooting step.

3. Answer all future questions using the same customer support persona.

Expected Output

- Consistent role-based responses
- Maintained tone and behavior across turns

6. Result

After completing this exercise, students are able to design persona-based prompts that guide a Large Language Model to respond according to defined roles. The exercise demonstrates how persona definition improves realism, consistency, and effectiveness in interview and support-based interactions.

8. Evaluate Prompt Performance Using Relevance, Coherence, and Correctness

1. Aim

To evaluate the performance of prompts by analyzing the relevance, coherence, and correctness of responses generated by a Large Language Model (LLM).

2. Objectives

- To understand key criteria for evaluating LLM responses
- To analyze prompt effectiveness based on relevance
- To assess coherence and logical flow in generated responses
- To verify the correctness and accuracy of outputs

3. Problem Statement

Even well-written prompts may sometimes generate responses that are irrelevant, incoherent, or incorrect. Without systematic evaluation, it is difficult to judge the quality of prompt outputs. This exercise focuses on evaluating prompt performance using standard qualitative criteria to ensure reliable and meaningful LLM responses.

4. Theory / Background

Prompt evaluation is an essential step in prompt engineering. It involves assessing the quality of model responses based on specific metrics:

- **Relevance**: How closely the response addresses the given prompt
- **Coherence**: Logical flow, clarity, and organization of ideas
- **Correctness**: Factual accuracy and correctness of information

By evaluating responses using these criteria, prompts can be refined to improve output quality and consistency.

5. Procedure

Step 1: Evaluating Relevance

Sample Prompts

1. Explain Artificial Intelligence and its applications in healthcare.

2. List the advantages of cloud computing.

3. Describe the role of data science in business decision-making.

Evaluation Criteria

- Response directly addresses the prompt
- No unnecessary or unrelated information

Expected Output

- Relevant and focused explanation
- Prompt-related content only

Step 2: Evaluating Coherence

Sample Prompts

1. Explain Machine Learning with proper flow and structure.

2. Describe the process of software development step by step.

3. Explain cloud computing using headings and subpoints.

Evaluation Criteria

- Logical sequence of ideas
- Clear structure and readability

Expected Output

- Well-organized response
- Smooth transition between points

Step 3: Evaluating Correctness

Sample Prompts

```
1. Explain what overfitting means in machine learning.

2. Define supervised learning with an example.

3. State the capital of Australia.
```

Evaluation Criteria

- Factual accuracy
- Conceptual correctness

Expected Output

- Correct definitions and facts
- Accurate examples

Step 4: Comparative Prompt Evaluation

Sample Prompts

```
1. Explain Artificial Intelligence.
```

```
2. Explain Artificial Intelligence in simple terms with examples.

3. Explain Artificial Intelligence for a beginner in 5 bullet points.
```

Evaluation Criteria

- Compare outputs based on relevance, coherence, and correctness
- Identify which prompt produces the best response

Expected Output

- Identification of the most effective prompt
- Clear justification based on evaluation criteria

6. Result

After completing this exercise, we are able to evaluate prompt performance using relevance, coherence, and correctness as key criteria. The exercise demonstrates how systematic evaluation helps in identifying prompt weaknesses and improving the overall quality of responses generated by a Large Language Model.