

Experiment No.: 1

Title: Implementation of Basic Prompting Techniques for Large Language Models (LLMs)

1. Aim

To implement and analyze **basic prompting techniques** used with Large Language Models (LLMs) and study how different prompt structures influence the quality, accuracy, and relevance of model outputs.

2. Objectives

After completing this experiment, the learner will be able to:

1. Understand the concept of prompting in LLMs
 2. Implement zero-shot, one-shot, and few-shot prompting
 3. Apply instruction-based and role-based prompting
 4. Compare outputs generated by different prompting strategies
 5. Analyze the effectiveness of prompt engineering techniques
-

3. Software & Hardware Requirements

Software Requirements

- Python 3.8 or above
- Hugging Face transformers library
- PyTorch

Hardware Requirements

- Minimum 8 GB RAM
 - Processor: Intel i5 / Ryzen 5 or higher
 - Internet connection (for first-time model download)
-

4. Theory

4.1 Large Language Models (LLMs)

Large Language Models are deep learning models trained on massive text datasets to understand and generate human-like language. Examples include GPT, BERT, and LLaMA.

LLMs work by predicting the **next word/token** based on the given context.

4.2 What is Prompting?

A **prompt** is an input text given to an LLM to instruct it to perform a specific task.

Prompting controls:

- Task behavior
 - Output format
 - Reasoning style
 - Context understanding
-

4.3 Importance of Prompting

- Eliminates the need for model retraining
 - Improves output accuracy
 - Reduces hallucinations
 - Enables task customization
-

5. Types of Basic Prompting Techniques

5.1 Zero-Shot Prompting

The model performs a task **without seeing any example**.

Use case: General questions, definitions

5.2 One-Shot Prompting

The model is given **one example** before performing the task.

Use case: Simple pattern learning

5.3 Few-Shot Prompting

The model is provided **multiple examples**.

Use case: Translation, classification, formatting

5.4 Instruction-Based Prompting

Clear and direct instructions guide the output.

Use case: Summarization, explanation, reasoning

5.5 Role-Based Prompting

The model is assigned a role or persona.

Use case: Teaching, expert explanations, domain-specific tasks

6. Algorithm

1. Install required libraries
 2. Load a pre-trained LLM
 3. Define prompts for each prompting technique
 4. Generate responses from the model
 5. Observe and compare outputs
 6. Analyze the effectiveness of each technique
-

7. Detailed Implementation (Python)

7.1 Install Required Libraries

pip install transformers torch

7.2 Import Libraries

```
from transformers import pipeline
```

7.3 Load Pre-Trained Model

```
text_generator = pipeline(  
    task="text-generation",  
    model="gpt2",  
    max_length=150  
)
```

7.4 Zero-Shot Prompting Implementation

```
prompt = "Explain Artificial Intelligence in simple words."
```

```
output = text_generator(prompt)  
print("Zero-Shot Output:\n", output[0]["generated_text"])
```

7.5 One-Shot Prompting Implementation

```
prompt = """"
```

Q: What is a stack?

A: A stack is a linear data structure that follows LIFO.

Q: What is a queue?

A:

""""

```
output = text_generator(prompt)  
print("One-Shot Output:\n", output[0]["generated_text"])
```

7.6 Few-Shot Prompting Implementation

prompt = "...."

English: Apple

French: Pomme

English: Book

French: Livre

English: Computer

French:

.....

```
output = text_generator(prompt)  
print("Few-Shot Output:\n", output[0]["generated_text"])
```

7.7 Instruction-Based Prompting Implementation

prompt = "Summarize the importance of databases in exactly 3 lines."

```
output = text_generator(prompt)  
print("Instruction-Based Output:\n", output[0]["generated_text"])
```

7.8 Role-Based Prompting Implementation

prompt = "...."

You are a senior computer science professor.

Explain the concept of recursion with a real-life example.

.....

```
output = text_generator(prompt)
print("Role-Based Output:\n", output[0]["generated_text"])
```

8. Sample Output (Expected)

- **Zero-Shot:** General explanation with moderate clarity
- **One-Shot:** More structured and relevant output
- **Few-Shot:** Accurate and pattern-based output
- **Instruction-Based:** Concise and controlled response
- **Role-Based:** Detailed, expert-level explanation