Tab 1

ChatGPT: Overview and Applications

# **Origin and Background**

ChatGPT is an advanced AI chatbot developed by OpenAI, based on the GPT (Generative Pre-trained Transformer) architecture. The GPT model uses transformer neural networks, introduced by Google in 2017, which allow it to understand and generate human-like text.

# **How ChatGPT Works**

ChatGPT is trained on massive datasets from books, websites, and other texts using unsupervised learning and reinforcement learning from human feedback (RLHF). The core mechanism behind its understanding is the Transformer architecture, which uses mechanisms like self-attention to weigh the importance of words in a sentence relative to others. This allows ChatGPT to generate relevant, context-aware responses. Users input a prompt, and ChatGPT predicts the next best word/token, generating coherent responses.

# **Real-World Applications of ChatGPT**

## **1. Education**

- Acts as a tutor for subjects like math, science, and language learning.  
- Assists in writing essays, generating summaries, and providing study tips.

## **2. Healthcare**

- Used in mental health chatbots for initial counseling and emotional support.  
- Helps doctors by summarizing patient reports or generating medical documentation.

## **3. Business**

- Automates customer support and query handling.  
- Assists in content creation, market research, and data analysis.

## **4. Creativity**

- Helps generate stories, poems, music lyrics, and script ideas.  
- Assists designers and artists in brainstorming concepts and refining ideas.

# **Conclusion**

ChatGPT represents a powerful shift in how humans interact with machines. From learning and working to creating and exploring, its real-world applications are vast. As technology evolves, ChatGPT continues to play a significant role in shaping the future of communication and automation.

Types of Prompts and Prompt Engineering Basics

# **Objective:**

To understand and apply different types of prompts: instructional, interrogative, zero-shot, and few-shot, and explore how prompt engineering improves AI responses.

# **1. Instructional Prompts**

These give direct instructions or commands to the AI.

| Prompt | Response (summary) |
| --- | --- |
| Summarize the story of \*Cinderella\*. | Gave a concise summary including main events. |
| Write a poem about space in four lines. | Delivered a short rhyming poem themed on space. |
| Convert this sentence to passive voice: "The dog chased the cat." | "The cat was chased by the dog." |
| Explain Newton's First Law in simple words. | Simple, student-friendly explanation. |
| Translate "Good Morning" into French. | "Bonjour." |

Result: Instructional prompts work well for straightforward tasks; responses were clear and followed the command accurately.

# **2. Interrogative Prompts**

These are framed as questions, inviting informative answers.

| Prompt | Response (summary) |
| --- | --- |
| What causes a rainbow? | Explained light refraction, reflection, and dispersion. |
| How do volcanoes erupt? | Step-by-step explanation with geological terms. |
| Why is the sky blue? | Described Rayleigh scattering effectively. |
| What is the capital of Japan? | "Tokyo." |
| Who invented the telephone? | "Alexander Graham Bell." |

Result: Interrogative prompts are effective for factual and explanatory answers. They allow the AI to expand in an informative tone.

# **3. Zero-shot Prompts**

No example is given — the AI relies solely on the prompt.

| Prompt | Response (summary) |
| --- | --- |
| Write a haiku about winter. | Created a haiku (5-7-5 syllable structure). |
| Describe a cat in one sentence. | Gave a vivid one-line description. |
| Make a list of five vegetables. | Provided a list without extra formatting. |
| Explain photosynthesis. | Detailed answer with key stages. |
| Write a code to reverse a string in Python. | Outputted correct code. |

Result: AI performs well with general knowledge, but output may lack consistency in format without examples.

# **4. Few-shot Prompts**

These include examples before the task.

Prompt Example:  
Example 1: Input: Translate "Hello" to Spanish. Output: "Hola"  
Example 2: Input: Translate "Thank you" to Spanish. Output: "Gracias"  
Now, Input: Translate "Good Night" to Spanish. Output:  
Response: "Buenas noches"

| Context Given | Prompt | Output |
| --- | --- | --- |
| Two math examples | Solve 8+5 | "13" with steps |
| Two jokes | Tell a joke | Returned joke in same style |
| Two resume bullet points | Add one for "Team Management" | Mimicked tone and format |
| Two Python list examples | Create a list of animals | Correct format used |
| Two product reviews | Write one for a phone | Review matched tone and style |

Result: Few-shot prompting improved accuracy, structure, and tone, especially for format-dependent tasks.

# **Prompt Engineering Insights**

- Context boosts relevance: Few-shot examples guide the model.  
- Clarity matters: Direct, clear instructions reduce ambiguity.  
- Formatting helps: Specifying output structure leads to better results.  
- Complexity influences output: More complex prompts yield richer, more detailed answers.

# **Conclusion**

Understanding and applying different prompt types improves AI output significantly. Zero-shot works well for general tasks, but few-shot prompting enhances formatting, tone, and task-specific consistency. Instructional and interrogative styles are both useful — the former for tasks, the latter for information.

Lab3: Precision Prompting for Information Extraction

# **Objective:**

Extract structured data from ChatGPT responses using precision prompts.

# **Activities:**

## **1. Ask ChatGPT to format its response in JSON or tabular format**

Prompt:  
"List the some planets in the solar system along with their average distance from the sun in JSON format."

Output:

[  
 {"planet": "Mercury", "distance\_from\_sun\_km": 57900000},  
 {"planet": "Venus", "distance\_from\_sun\_km": 108200000},  
 {"planet": "Earth", "distance\_from\_sun\_km": 149600000},  
]

Accuracy: 100% correct.

Format: Perfectly structured JSON.

## **2. Extract contact info from a paragraph**

Prompt:  
"Extract name, phone number, and email from the following text:  
  
'Hi, I'm Priya Sharma. You can reach me at priya.sharma@example.com or call me on 9876543210 for more details.'"

Output:

{  
 "name": "Priya Sharma",  
 "phone": "9876543210",  
 "email": "priya.sharma@example.com"  
}

Accuracy: 100% accurate.

Format: JSON structure is ideal for database input.

## **3. Extract resume sections from a paragraph**

Prompt:  
"Identify the following sections from this resume snippet: education, skills, and experience.  
  
'The candidate completed a B.Tech in Computer Science in 2020. Skilled in Java, Python, and SQL. Worked at Infosys as a Software Engineer from 2021 to 2023.'"

Output:

{  
 "education": "B.Tech in Computer Science, 2020",  
 "skills": ["Java", "Python", "SQL"],  
 "experience": "Software Engineer at Infosys (2021-2023)"  
}

Accuracy: High, structured appropriately.

Format: JSON suitable for automated parsing.

## **4. Extract keywords from a paragraph**

Prompt:  
"List 5 important keywords from the following paragraph:  
  
'Artificial Intelligence is transforming industries by enabling automation, enhancing decision-making, and personalizing customer experiences.'"

Output:

[  
 "Artificial Intelligence",  
 "automation",  
 "decision-making",  
 "personalizing",  
 "customer experiences"  
]

Accuracy: 100% relevant keywords.

Format: Clean JSON list format.

**Conclusion:**

Precision prompting significantly improves data extraction quality from AI responses. By clearly specifying structure (like JSON or tables), we can obtain highly usable outputs suitable for downstream automation, analysis, or integration with applications.

## **Lab 4: Summarization and Text Transformation**

**Objective:** Use ChatGPT to summarize content and transform text tone.

### **Activity 1: Summarize News Article**

**Prompt Used:**

Summarize the following news article in 3 sentences:  
 "India’s GDP grew 7.8% in Q1 FY25, driven by robust manufacturing and services sector growth..."

**Summary Output:** India’s GDP grew by 7.8% in Q1 FY25, led by strong performance in manufacturing and services. Private consumption and government spending contributed significantly. Economists expect steady growth, although global risks remain.

### **Activity 2: Email Tone Transformation**

**Prompt Used:**

Rewrite the following email in a formal tone:  
 “Hey! Just checking if you saw my last email. Can you send the report by tomorrow?”

**Original Email (Informal):** “Hey! Just checking if you saw my last email. Can you send the report by tomorrow?”

**Transformed Email (Formal):** “Dear [Name],  
 I hope this message finds you well. I wanted to follow up on my previous email and kindly request the report by tomorrow, if possible.  
 Best regards,  
 [Your Name]”

### **Evaluation Summary:**

| **Task** | **Accuracy** | **Tone/Style** | **Clarity** |
| --- | --- | --- | --- |
| News Summarization | ✅ | ✅ Concise | ✅ |
| Email Transformation | ✅ | ✅ Formal | ✅ |

### **Conclusion:**

ChatGPT effectively summarized content and adjusted tone according to context. Both outputs were accurate, concise, and appropriate for their respective tasks.

## **Lab 5: ChatGPT in Code Generation and Debugging**

**Activity Chosen:** Python Code Generation  
 **Objective:** Explore how ChatGPT can assist in programming tasks like code writing and debugging.

### **Prompt Used:**

Write a Python program to check if a number is prime.

### **ChatGPT’s Generated Code:**

python

CopyEdit

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5)+1):

if n % i == 0:

return False

return True

# Example usage

num = 29

if is\_prime(num):

print(f"{num} is a prime number.")

else:

print(f"{num} is not a prime number.")

### **Expected Output:**

csharp

CopyEdit

29 is a prime number.

### **Evaluation:**

| **Criteria** | **Result** |
| --- | --- |
| **Correctness** | ✅ Matches expected logic & output |
| **Code Quality** | ✅ Efficient, clear, and well-structured |
| **Edge Cases** | ✅ Handles 0, 1, and negative inputs |
| **Language Used** | ✅ Python |

### **Conclusion:**

ChatGPT successfully generated a correct and optimized Python function to identify prime numbers. The output matched expectations, showing ChatGPT's practical use in beginner-level code generation.

## **Lab 6: Domain-Specific Applications**

**Activity Chosen:** Educational Quiz Generation  
 **Objective:** Use ChatGPT to perform a domain-specific task in the education sector.

### **Prompt Used:**

Generate a 5-question multiple choice quiz on the topic "Basics of Financial Markets" suitable for undergraduate commerce or finance students. Each question should have 4 options and indicate the correct answer.

### **Task Performed:**

ChatGPT generated a structured, 5-question multiple choice quiz focused on fundamental financial market concepts. Each question included 4 options and clearly marked the correct answer.

### **Evaluation Summary:**

| **Criteria** | **Evaluation Summary** |
| --- | --- |
| **Domain Fit** | ✅ Finance-focused, education appropriate |
| **Clarity** | ✅ Clear, concise, and understandable |
| **Accuracy** | ✅ All concepts and answers factually correct |
| **Use Case** | ✅ Ready-to-use for classroom or self-study |
| **Format** | ✅ Well-structured, MCQ with answers marked |

### **Conclusion:**

This activity demonstrates how ChatGPT can generate accurate, structured, and relevant educational content for domain-specific learning. The quiz is suitable for undergraduate-level commerce/finance students.