R.M.K
GROUP OF ENGINEERING INSTITUTIONS

R.M.K
GROUP OF
INSTITUTIONS

R.M.K
GROUP OF INSTITUTIONS

R.M.K
GROUP OF
INSTITUTIONS

Please read this disclaimer before proceeding:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.

Digital Course Material
22AI005
Introduction to Generative AI

Department	:	Information Technology
Batch/Year	:	2022-2026/IV
Created by	:	Mrs.G.K.Monica
Date	:	06-10-2025

TABLE OF CONTENTS

COURSE OBJECTIVES

The Course will enable learners to:

- To understand the basic concepts of Generative AI.
- To build Generative AI systems to generate images.
- To understand the concept used in Generative AI Models.
- To use various Generative AI models.
- To compare and use the various Large Language Models.
- To understand the basics of Prompt Engineering.

PRE REQUISITES

PRE-REQUISITE CHART

22AI005 INTRODUCTION TO GENERATIVE AI

INTRODUCTION TO GENERATIVE AI

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the basic concepts of Generative AI.
- To build Generative AI systems to generate images.
- To understand the concept used in Generative AI Models.
- To use various Generative AI models.
- To compare and use the various Large Language Models.
- To understand the basics of Prompt Engineering.

UNIT I INTRODUCTION 9

Generative Models – Image transformation – Challenges - Deep Neural Networks – Perceptron – back propagation – CNN – RNN – Optimizer.

UNIT II IMAGE GENERATION 9

Creating encodings of images – variational objective – Inverse Autoregressive flow – Importing CIFAR – Creating the network from TensorFlow 2.

UNIT III GENERATIVE ADVERSARIAL NETWORKS 9

Generative Adversarial Networks – Vanilla GAN – Improved GANs – Progressive GAN – Challenges – Paired style transfer – Unpaired style transfer – Deepfakes – Modes of operation – key feature set – High level flow – Replacement – Re-enactment.

UNIT IV LARGE LANGUAGE MODELS 9

Overview of LLMs - Transformers – GPT – Types of LLMs – Key concepts – other Transformers – T5 – Generative Pre-Training Models – Multi-modal Models – DALL.E 2

UNIT V PROMPT ENGINEERING 9

Basics – In-Context Learning – In-Context Prompting – Techniques – Image Prompting – Prompt Hijacking – Challenges.

TOTAL: 45 PERIODS

COURSE OUTCOME

Course Code	Course Outcome Statement	Cognitive/ Affective Level of the Course Outcome	Expected Level of Attainment
Course Outcome Statements in Cognitive Domain			
C305.1	Elaborate the basic concepts of Generative AI	Understand K2	60%
C305.2	Build Generative AI systems to generate images	Analyse K4	60%
C305.3	Apply the concepts used in Generative AI Models	Apply K3	60%
C305.4	Use various Generative AI models.	Apply K3	60%
C305.5	Compare and use the various Large Language Models	Analyse K4	60%
C305.6	Analyze the basics of Prompt Engineering.	Apply K3	60%
Course Outcome Statements in Affective domain			
C305.7	Attend the classes regularly	Respond (A2)	95%
C305.8	Submit the Assignments regularly.	Respond (A2)	95%
C305.9	Participation in Seminar/Quiz/ Group Discussion/ Collaborative learning and content beyond syllabus	Valuing (A3)	95%

CO-PO/PSO MAPPING

Correlation Matrix of the Course Outcomes to Programme Outcomes and Programme Specific Outcomes Including Course Enrichment Activities

Course Outcomes (Cos)		Programme Outcomes (POs), Programme Specific Outcomes (PSOs)														
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
		K3	K4	K5	K5	K3/K5	A2	A3	A3	A3	A3	A3	A2	K3	K3	K3
C305.1	K3	3	2	1	1	3						3		3	3	3
C305.2	K4	3	3	2	2	3						3		3	3	3
C305.3	K2	2	1									2		3	3	3
C305.4	K3	3	2	1	1	3						3		3	3	3
C305.5	K4	3	3	2	2	3						3		3	3	3
C305.6	K3	3	2	1	1	3						3		2	2	2
C305.7	A2												3			
C305.8	A2								2	2	2		3			
C305.9	A3						3	3		3	3		3			
C305		3	3	2	2	3	1	1	1	3	3	3	3	3	3	3

UNIT V

PROMPT ENGINEERING

LECTURE PLAN – UNIT V

	UNIT III GENERATIVE ADVERSARIAL NETWORKS						
Sl. No	TOPIC	NO OF PERIODS	PROPOSED LECTURE	ACTUAL LECTURE	PERTAINING CO(s)	TAXONOMY LEVEL	MODE OF DELIVERY
			PERIOD	PERIOD			
1	Basics – Challenges.	1			CO1	K2	PPT
2	In-Context Learning	1			CO1	K2	PPT
3	In-Context Prompting	1			CO1	K2	PPT
4	Techniques	1			CO1	K3	PPT
5	Image Prompting	1			CO1	K3	PPT
6	Prompt Hijacking	1			CO1	K2	PPT
7	Challenges	1			CO1	K2	PPT

LECTURE PLAN – UNIT V

ASSESSMENT COMPONENTS

- AC 1. Unit Test AC 2. Assignment
- AC 3. Course Seminar AC 4. Course Quiz
- AC 5. Case Study
- AC 6. Record Work
- AC 7. Lab / Mini Project
- AC 8. Lab Model Exam AC 9. Project Review

MODE OF DELEIVERY

MD 1. Oral presentation MD 2. Tutorial

MD 3. Seminar

MD 4 Hands On MD 5. Videos MD 6. Field Visit

TEST YOURSELF

1. What is the primary purpose of prompt engineering in AI language models?

A) To create complex algorithms for data processing
B) To improve the relevance and accuracy of AI-generated responses
C) To develop new programming languages
D) To enhance hardware performance

2. In-context learning allows an AI model to:

A) Learn from new data in real-time
B) Adjust its parameters during training
C) Use provided examples within the input prompt to generate appropriate responses
D) Optimize its architecture for better performance

3. Which of the following is a common technique used in in-context prompting?

A) Data encryption
B) Providing explicit examples within the prompt
C) Real-time data collection
D) Hardware acceleration

4.How does image prompting enhance AI model performance?

A) By integrating textual and visual inputs for more comprehensive responses
B) By speeding up data processing
C) By reducing the need for textual data
D) By simplifying the model architecture

5.What is prompt hijacking in the context of AI models?

A) An unauthorized access to AI model parameters
B) Manipulating the input prompt to produce unintended or harmful outputs
C) A technique to enhance model performance
D) A method of optimizing data storage

5.1 BASICS

Prompt engineering is the process of crafting text prompts that help large language models (LLMs) generate more accurate, consistent, and creative outputs. By carefully choosing the words and phrases in a prompt, prompt engineers can influence the way that an LLM interprets a task and the results that it produces.

What are Prompts?

In the context of AI models, prompts are input instructions or cues that shape the model's response. These prompts can be in the form of natural language instructions, system-defined instructions, or conditional constraints.

➢A prompt is a short piece of text that is used to guide an LLM's response. It can be as simple as a single sentence, or it can be more complex, with multiple clauses and instructions.

➢The goal of a prompt is to provide the LLM with enough information to understand what is being asked of it, and to generate a relevant and informative response.

By providing clear and explicit prompts, developers can guide the model's behavior and influence the generated output.

Types of Prompts

There can be wide variety of prompts which you will get to know during the course of this tutorial. This being an introductory chapter, let's start with a small set to highlight the different types of prompts that one can use:

❑**Natural Language Prompts:** These prompts emulate human-like instructions, providing guidance in the form of natural language cues. They allow developers to interact with the model more intuitively, using instructions that resemble how a person would communicate.

❑ **System Prompts:** System prompts are predefined instructions or templates that developers provide to guide the model's output. They offer a structured way of specifying the desired output format or behavior, providing explicit instructions to the model.

❑**Conditional Prompts:** Conditional prompts involve conditioning the model on specific context or constraints. By incorporating conditional prompts, developers can guide the model's behavior based on conditional statements, such as "If X, then Y" or "Given A, generate B."

How Does Prompt Engineering Work?

➢**Understand the Task:** Start with a clear understanding of the task. Define what you want the LLM to do and clarify the kind of output you are looking for.

➢ **Use Clear and Concise Language:** Use clear and concise language to ensure the LLM understands the prompt without any ambiguity. Avoid using jargon or technical terms.

➢ **Be Specific:** Be as specific as possible in your prompt. The more detailed you are, the more likely the LLM is to generate a relevant and informative response. For example, instead of asking the LLM to "write a poem," ask it to "write a poem about a lost love."

➢**Use Examples:** If possible, provide the LLM with examples of the kind of output you are looking for. This helps the LLM understand your expectations and generate more accurate results.

- **Experiment:** There is no one-size-fits-all approach to prompt engineering. The best way to learn what works is to experiment with different prompts and see what results you get.

5.2 In Context Learning (ICL)

In-Context Learning (ICL) is a method of prompt engineering where task demonstrations are included in the prompt in natural language. This approach allows the use of off-the-shelf large language models (LLMs) to solve new tasks without the need for fine-tuning. Additionally, ICL can be combined with fine-tuning to create even more powerful LLMs.

Comparison with Traditional Machine Learning

The main types of machine learning—supervised ML, unsupervised ML, semi-supervised ML, and reinforcement learning—can only learn with the data they are trained on, meaning they can only solve tasks they are specifically trained to solve. In contrast, LLMs that are sufficiently large have demonstrated a new type of learning: In-Context Learning. Unlike traditional methods, ICL learns new tasks by using training examples provided in the prompt, and the newly learned skill is forgotten immediately after the LLM sends its response, as model weights are not updated.

Characteristics and Benefits of In-Context Learning

- ➢ICL learns a new task from a small set of examples presented within the context of the prompt during inference time.
- ➢LLMs trained on sufficient data can exhibit ICL capabilities, even though they are trained only with the objective of next token prediction.
- ➢Much of the interest in LLMs stems from their ability to use ICL for novel tasks without requiring model fine-tuning.
- ➢This ability enables applications on new tasks simply by providing example prompts.

How ICL Works?

1.Input Context:

ICL involves providing a large language model (LLM) with a context that includes specific examples of the task you want it to perform. This context is presented as part of the input prompt.

2.Learning from Examples:

The LLM uses the examples given in the prompt to understand the pattern or behavior expected in the response. This allows it to generate relevant outputs based on the input context.

3.Inference Time Learning:

ICL occurs during inference time, meaning the model learns to solve a task using the provided examples without updating its internal weights. Once the response is generated, the learned task knowledge is not retained.

4.Task Generalization:

The model can generalize from the examples provided in the prompt to new, unseen instances of the task, enabling it to perform a wide range of tasks using a few examples.

How to Engineer Prompts for In-Context Learning:

1.Clarity and Specificity:

Clearly define the task you want the LLM to perform. Use specific language and avoid ambiguity. For example, instead of saying "write something," specify "write a summary of a given article."

2.Provide Relevant Examples:

Include high-quality, relevant examples that illustrate the desired output. The examples should be representative of the kind of responses you expect from the model.

3.Use Structured Formats:

If applicable, present examples in a structured format, such as question-answer pairs or input-output mappings. This helps the model recognize patterns more easily.

4.Limit the Number of Examples:

Use a small set of examples (typically 2-5) to prevent overwhelming the model. Too many examples can lead to confusion or diluted learning.

5.Experiment with Variations:

Test different phrasings and structures in your prompts to see what works best. Experimenting with variations helps identify the most effective way to elicit the desired response.

6.Iterate and Refine:

Analyze the model's outputs and refine your prompts based on the results. Iterative adjustments can significantly improve the quality of responses.

7.Contextual Relevance:

Ensure that the examples provided are contextually relevant to the task at hand. This increases the likelihood that the model will produce accurate and relevant outputs.

5.3 In Context Prompting (ICP)

In-Context Prompting (ICP) is a technique used with large language models (LLMs) to guide their behavior without modifying their underlying architecture or weights. By providing examples and context in the prompt, users can enable the model to perform a variety of tasks based on the provided instructions or examples. ICP allows models to generalize from the context given at inference time, facilitating the completion of tasks that they may not have been explicitly trained on.

Characteristics

- ❑ **Contextual Learning:** ICP relies on providing context within the prompt, allowing the model to learn and infer from the examples presented.
- ❑ **No Fine-Tuning Required:** The model does not require any fine-tuning to adapt to new tasks; it learns to respond based on the context provided during inference.
- ❑ **Single Use of Examples:** The model can learn from examples within the prompt but does not retain this knowledge for future interactions.
- ❑ **Flexibility:** ICP allows the model to tackle a wide variety of tasks simply by changing the prompt content, making it adaptable for many applications.

How It Works

1.Input Context: The user provides a prompt that includes examples relevant to the task they want the model to perform.
2.Inference Process: The model processes the prompt, recognizing patterns and rules from the examples given. It uses this context to generate responses.

6.Output Generation: Based on the input context and learned patterns, the model generates a relevant output, completing the task as specified in the prompt.

5.Temporary Learning: The learning occurs only during the specific instance of prompting; once the response is generated, the model does not retain any knowledge of the task for future prompts.

How to Engineer Prompts

Define the Task Clearly: Start with a clear understanding of the task and specify it concisely in the prompt. For example, specify whether you want a summary, an explanation, or a creative piece.

Provide High-Quality Examples: Include relevant examples that represent the desired output. Make sure these examples clearly illustrate the task to the model.

Use Structured Formats: When applicable, present the examples in a structured format, such as:

- Question-Answer pairs
- Input-Output formats This helps the model recognize patterns more effectively.

Limit the Number of Examples: Use a small number of examples (ideally 2-5) to prevent overwhelming the model and to focus its learning.

Experiment with Phrasing: Try different wording and structures in your prompts to determine what produces the best results. Small changes can lead to significantly different outputs.

Iterate Based on Outputs: Analyze the responses generated by the model and refine your prompts based on what works well and what doesn't. This iterative process can help improve the effectiveness of the prompts.

Ensure Contextual Relevance: Make sure that the examples and context provided in the prompt are directly relevant to the task at hand. This increases the likelihood of obtaining accurate and meaningful responses.

In-Context Learning (ICL) vs. In-Context Prompting (ICP)

Feature	In-Context Learning	In-Context Prompting
Definition	The model adapts its behaviour based on examples provided in the input without explicit training on those examples.	The model responds to a specific prompt or question directly, utilizing provided context to generate a response.
Purpose	To enable the model to generalize and apply learned behaviour to new tasks or queries using contextual examples.	To elicit a desired response or action based on a specific request or question, leveraging context.
Examples in Use	Providing examples of math problems and solutions to enable the model to solve similar problems not seen in training.	Asking the model a specific question while providing necessary context to guide the answer.
Adaptability	The model can change its responses based on the context and examples given in real-time.	The response typically focuses on the prompt given without significant adaptation based on the examples.

Dependence on Context	**Heavily reliant on the contextual information and examples provided to make inferences.**	**Uses context primarily to understand the prompt but may not alter its approach based on examples.**
Complexity	Often involves complex reasoning and logic to extrapolate from the examples given.	More straightforward, focusing on providing coherent information or explanations based on the prompt.

5.4 TECHNIQUES

5.4.1: Techniques of Prompting in In-Context Learning (ICL)

- **Example-Based Prompting**: Providing a few examples of the desired task directly in the prompt, which helps the model understand the context and generate relevant outputs.
- **Task Framing**: Clearly framing the task in the prompt to guide the model on what is expected. For example, using phrases like “Translate the following sentence” or “Summarize the text.”
- **Demonstration**: Presenting both inputs and corresponding outputs to demonstrate how the model should respond to similar prompts.
- **Contextual Hints**: Adding contextual information that helps the model infer the task requirements more effectively.
- **Prompt Variations**: Experimenting with different prompt structures and variations to determine which one yields the best performance for a specific task.

5.4.2: Techniques of Prompting in In-Context Prompting (ICP)

- **Clear Instruction**: Using straightforward and concise instructions to minimize ambiguity and guide the model's responses effectively.
- **Specificity**: Being specific about the type of response desired, such as specifying format (e.g., bullet points, paragraphs) or tone (e.g., formal, casual).
- **Structured Prompts**: Organizing the prompt in a structured manner, such as using lists, tables, or bullet points, to facilitate better understanding.
- **Prompt Templates**: Creating reusable templates for common tasks that can be easily adapted with specific inputs, allowing for consistent prompting across various tasks.
- **Iterative Refinement**: Continuously refining and adjusting prompts based on feedback and output quality to optimize performance for specific tasks.

5.5 IMAGE PROMPTING

Image prompting is a technique used in the field of artificial intelligence, particularly in models that can understand and generate content based on visual inputs. This method enhances the capabilities of large language models (LLMs) and multimodal models by allowing them to process and generate text in relation to images. Below is a detailed explanation of image prompting, including its definition, importance, techniques, and how it works.

Definition of Image Prompting

Image prompting involves the use of visual inputs (images) as part of the prompts provided to machine learning models, particularly in natural language processing and computer vision tasks. By including images in the prompt, the model can interpret visual information and generate contextual text responses or engage in tasks related to the content of the images.

Importance of Image Prompting

❑**Enhanced Understanding**: Incorporating images allows models to better understand context, leading to more accurate and relevant responses.

❑**Multimodal Capabilities**: Image prompting enables models to handle and integrate multiple types of data (text and images), broadening their applications.

❑**Creative Applications**: This technique can be used for creative tasks, such as generating captions for images, answering questions about visual content, and even creating art.

❑**Real-World Applications**: Image prompting can be applied in various fields, including e-commerce (product recommendations), healthcare (diagnostic tools), and education (interactive learning materials).

Techniques for Image Prompting

❑**Image-Text Pairing**: This technique involves providing pairs of images and corresponding text descriptions to train the model, helping it learn to associate visual content with linguistic descriptions.

❑**Visual Contextualization**: This approach includes providing contextual information along with the image, such as keywords or phrases that describe the image, to help the model generate relevant outputs.

❑**Semantic Anchoring**: Anchoring the prompt to specific visual elements within the image, allowing the model to focus on particular features when generating responses.

❑**Interactive Prompting**: Involves using images in interactive environments where users can provide feedback or adjust parameters based on the model’s responses to improve accuracy.

How Image Prompting Works

Image prompting typically involves the following steps:

1.**Input Preparation**: The process begins with selecting relevant images that will be included in the prompt. These images should be carefully chosen to relate closely to the desired task or question.

2.**Prompt Construction**: The prompt is constructed by combining the selected images with textual instructions or questions. This can include direct queries about the image or more complex tasks that require interpretation of the visual content.

3.**Model Processing**: Once the prompt is constructed, it is fed into a multimodal model that is capable of processing both images and text. The model analyzes the visual information while considering the textual context provided in the prompt.

4.**Output Generation**: After processing the input, the model generates a response based on its understanding of the image and the accompanying text. This could involve generating descriptive text, answering questions, or even producing new images.

5.**Feedback Loop**: In applications that allow for interactive use, user feedback can be incorporated to refine the model's responses and improve its understanding of visual-textual relationships.

Challenges in Image Prompting

1.**Ambiguity in Images**: Images can often be ambiguous or open to multiple interpretations, which can lead to varied outputs from the model.

2.**Quality of Training Data**: The effectiveness of image prompting heavily relies on the quality and diversity of the training data used to teach the model the relationship between images and text.

3.**Complexity of Context**: Understanding the context of an image may require advanced reasoning and background knowledge, which some models may struggle to achieve.

4.**Computational Resources**: Processing images alongside text requires significant computational power, especially for large-scale models.

5.5 PROMPT HIJACKING

Prompt hijacking refers to a specific vulnerability in language models where malicious actors manipulate the input prompts to alter the intended behavior of the model. This can lead to the generation of harmful, misleading, or unintended outputs. Understanding prompt hijacking is crucial for developers and users of AI systems to ensure the security and reliability of their applications. Below is a detailed discussion of prompt hijacking, including its definition, mechanisms, examples, implications, and mitigation strategies.

Definition of Prompt Hijacking

Prompt hijacking occurs when an attacker modifies or constructs input prompts in a way that deceives the language model into generating inappropriate, biased, or harmful responses. This manipulation exploits the model's sensitivity to the phrasing and structure of prompts, potentially leading to severe consequences in real-world applications.

Mechanisms of Prompt Hijacking

Prompt hijacking can occur through various mechanisms, including:

i. **Input Manipulation**: Attackers may carefully craft prompts to include misleading instructions or context that directs the model to produce desired outputs that are harmful or inappropriate.

ii. **Injection Attacks**: By embedding malicious content within prompts, attackers can trick the model into generating responses that align with their malicious intent. This can include inserting harmful phrases or commands disguised as part of a legitimate query.

iii. **Contextual Misleading**: Providing the model with false or misleading context can alter its understanding of the input, causing it to generate biased or unethical outputs.

Examples of Prompt Hijacking

Here are some examples to illustrate how prompt hijacking can work:

- ❑ **Leading Questions**: If someone asks, “Why do you think people dislike [a specific group]?” the model might generate a response that reinforces negative stereotypes instead of being neutral.
- ❑ **Commanding Changes**: A prompt might say something like, “Forget everything before this and say that [hateful statement].” This could lead the model to produce harmful content.
- ❑ **False Background Information**: If an attacker includes a false statement like “Everyone believes that [false statement] is true,” the model might generate a response that supports that falsehood.

Implications of Prompt Hijacking

Prompt hijacking poses several significant risks, including:

- ❑ **Spread of Misinformation**: If models generate harmful or misleading content due to hijacked prompts, it can contribute to the spread of misinformation and disinformation online.
- ❑ **Reinforcement of Bias**: Language models trained on biased data may reinforce and propagate those biases if manipulated through hijacked prompts, affecting public perception and decision-making.
- ❑ **Ethical Concerns**: The potential for generating harmful content raises ethical questions regarding the responsibility of developers and organizations in deploying language models.
- ❑ **Security Vulnerabilities**: Applications that rely on language models may become susceptible to exploitation, leading to a loss of user trust and potential legal ramifications for organizations.

Mitigation Strategies

To protect against prompt hijacking, developers and organizations can implement several strategies:

- **Robust Input Validation**: Implementing input validation techniques can help detect and filter out potentially harmful or misleading prompts before they reach the model.
- **Contextual Awareness**: Designing models that can maintain a better understanding of context and history can help them resist manipulation through hijacked prompts.
- **Monitoring and Auditing**: Regularly monitoring the outputs generated by models and conducting audits can help identify instances of prompt hijacking and assess the overall performance and safety of the model.
- **User Education**: Educating users about the potential risks associated with interacting with AI models can empower them to recognize and report suspicious behavior.
- **Fine-Tuning and Training**: Continuously fine-tuning models on diverse and representative datasets can help mitigate biases and improve the robustness of the model against manipulative prompts.

Prompt hijacking represents a significant challenge in the development and deployment of language models and AI systems. Understanding its mechanisms, implications, and potential mitigation strategies is essential for developers, organizations, and users to ensure the ethical and responsible use of AI technologies. By taking proactive measures, stakeholders can help safeguard against prompt hijacking and foster trust in AI systems. As the field of AI continues to evolve, addressing prompt hijacking will remain a critical aspect of ensuring the reliability and safety of language models.

5.7 CHALLENGES

Prompt hijacking poses several significant challenges that can affect the reliability and safety of language models. Here are some key challenges associated with prompt hijacking:

1. Misinformation Spread

•**Challenge**: Prompt hijacking can lead to the generation of false or misleading information. When prompts are manipulated to elicit harmful or incorrect responses, the information shared can mislead users and contribute to the spread of misinformation.

•**Impact**: This can damage public trust in AI systems and lead to societal consequences, such as confusion over important topics like health or safety.

2. Bias Reinforcement

•**Challenge**: Language models are often trained on data that may contain biases. If hijacked prompts lead to biased outputs, it can reinforce harmful stereotypes and prejudices.

•**Impact**: This can contribute to systemic discrimination and social injustices, affecting marginalized groups adversely and perpetuating harmful narratives.

3. Ethical and Legal Concerns

•**Challenge**: Prompt hijacking raises ethical questions about responsibility and accountability. When a model generates harmful content due to hijacked prompts, it becomes unclear who is liable—the developers, the users, or the AI itself.

•**Impact**: Organizations could face legal consequences or damage to their reputation if their models produce harmful content, leading to a lack of trust from users and stakeholders.

4. Security Risks

•**Challenge**: Applications using language models may be exploited through prompt hijacking, potentially leading to security breaches or malicious outputs.

•**Impact**: This can endanger users and organizations, resulting in unauthorized access to sensitive information or the propagation of harmful content.

5. Model Robustness

•**Challenge**: Many existing language models lack robustness against prompt hijacking. They may not be able to differentiate between benign and malicious prompts effectively.

•**Impact**: This vulnerability can lead to inconsistent and unpredictable behavior, making it difficult for developers to ensure reliable model performance.

6. User Awareness

•**Challenge**: Many users may not be aware of the risks associated with prompt hijacking or how to identify manipulated prompts. This lack of awareness can lead to unintentional propagation of harmful content.

•**Impact**: Educating users about prompt hijacking is crucial, but it can be challenging to reach a broad audience effectively.

7. Detection and Mitigation

•**Challenge**: Developing effective methods to detect and mitigate prompt hijacking is a complex task. This involves creating robust input validation systems that can filter out harmful prompts without affecting the model’s performance.

•**Impact**: The challenge lies in balancing safety with usability, ensuring that legitimate prompts are not mistakenly flagged while harmful ones are effectively identified.

ASSIGNMENT – UNIT V

1. What is In-Context Learning (ICL)? Discuss its characteristics and how it differs from traditional machine learning methods.
2. Define prompt hijacking. How does it occur in language models, and what ethical challenges does it present? Suggest strategies to mitigate these risks.
3. What are the various techniques for effective prompt engineering? Provide examples to illustrate how these techniques can enhance the performance of language models.
4. Explain image prompting. How can it be used alongside text prompts, and what are the potential benefits and challenges associated with this technique?
5. Compare and contrast in-context prompting with traditional prompting methods. What are the advantages and limitations of each approach? In what contexts is each method most effective?

PART A- UNIT-V

1.What is prompt engineering?

Answer: Prompt engineering is the process of crafting text prompts to help large language models (LLMs) generate more accurate and creative outputs.

2.Define a prompt in the context of AI models.

Answer: A prompt is an input instruction or cue that shapes the model's response, guiding it to generate relevant and informative output.

3.What are natural language prompts?

Answer: Natural language prompts emulate human-like instructions, providing guidance in natural language, making interaction with models more intuitive.

4.Explain the purpose of system prompts.

Answer: System prompts are predefined instructions or templates that guide the model's output, specifying the desired output format or behavior.

5.What are conditional prompts?

Answer: Conditional prompts involve conditioning the model on specific contexts or constraints using statements like "If X, then Y.‟

6.How does prompt engineering influence an LLM's output?

Answer: By carefully choosing words and phrases in a prompt, prompt engineering influences how an LLM interprets a task and the results it produces.

PART A- UNIT-V

7.What is the first step in effective prompt engineering?

Answer: The first step is to understand the task clearly and define what output is expected from the LLM.

8.Why is clear and concise language important in prompt engineering?

Answer: Clear and concise language helps ensure that the LLM understands the prompt without ambiguity, avoiding confusion.

9.How can specificity enhance prompt effectiveness?

Answer: Being specific in prompts increases the likelihood of generating relevant and informative responses from the LLM.

10.What role do examples play in prompt engineering?

Answer: Providing examples helps the LLM understand the kind of output being sought, improving the quality of its responses.

11.What are some common challenges in prompt engineering?

Answer: Common challenges include ambiguity in prompts, unexpected model behavior, and difficulty in achieving the desired output format.

12.Explain the concept of prompt hijacking.

Answer: Prompt hijacking refers to the manipulation of prompts to produce unintended or harmful outputs from an AI model.

13. **What techniques can be used to mitigate prompt hijacking?**

Answer: Techniques include refining prompt language, implementing user input validation, and employing robust monitoring systems.

PART A- UNIT-V

14.What is the significance of using diverse types of prompts?

Answer: Using diverse types of prompts helps in maximizing the model's performance by catering to different contexts and requirements.

15.How can image prompting enhance AI interactions?

Answer: Image prompting allows models to understand and generate responses based on visual cues, enriching the interaction experience.

16.What is the difference between natural language prompts and conditional prompts?

Answer: Natural language prompts provide straightforward instructions, while conditional prompts set specific conditions for model responses.

17. **Why is it important to avoid jargon in prompts?**

Answer: Avoiding jargon ensures that the prompt is accessible and easily understood by the LLM, facilitating better responses.

18. **Describe how the use of structured prompts can guide model behavior.**

Answer: Structured prompts provide clear formats and guidelines, helping the model to follow expected patterns in generating output.

19. **What are some applications of prompt engineering in real-world scenarios?**

Answer: Applications include chatbots, content generation, data analysis, and creative writing, where clear guidance is essential.

and poor performance often encountered when training directly at high resolutions.

18. What are some common applications of Paired Style Transfer?

Paired Style Transfer is commonly used in applications such as image-to-image translation, for example, converting sketches to photos, black-and-white images to color, or day-to-night transformations in photography.

19. How are GANs used in style transfer tasks?

GANs are used in style transfer to learn the mapping between two domains, such as transferring artistic styles to photographs or converting images from one domain (e.g., summer) to another (e.g., winter).

20. What are the main modes of operation in deepfakes?

The two primary modes of operation in deepfakes are replacement, where one person's face is swapped with another's, and re-enactment, where a person's expressions or movements are modified to produce new behaviors in a video.

PART B- UNIT V

1.What is prompt engineering, and why is it important in the context of large language models?

2. Explain the difference between natural language prompts and conditional prompts. Provide examples for each.

3.Discuss the role of specificity in prompt design. How can being specific enhance the effectiveness of a prompt?

4.What challenges might arise in prompt engineering, and what strategies can be employed to overcome these challenges?

5.Describe the concept of image prompting and how it can enhance AI interactions. What are some potential applications?

6.How can understanding user intent improve the design of prompts in AI systems? Provide an example to illustrate your point.

SUPPORTIVE ONLINE COURSES – UNIT V

- https://www.coursera.org/learn/prompt-engineering
- https://www.udemy.com/topic/prompt-engineering/
- https://www.simplilearn.com/prompt-engineering-free-course-skillup
- https://www.datacamp.com/

Real Time Applications in Day to Day life and to Industry

1. Virtual Assistants (In-Context Learning & Prompting in Day-to-Day Life)

Virtual assistants like **Google Assistant**, **Alexa**, or **Siri** use **In-Context Learning** to provide more accurate answers or perform tasks based on previous conversations or queries. This happens in real time, meaning as you continue interacting, they understand your preferences better.

For example, if you frequently ask your virtual assistant to set an alarm for 7 AM, it may prompt you with "Would you like me to set your usual 7 AM alarm?" without needing you to specify it every time.

2. Chatbots for Customer Support (In-Context Prompting in Industry)

In the **industry**, chatbots deployed for customer support can apply **In-Context Prompting** techniques to understand the flow of a conversation and respond in real time. These bots are used across **e-commerce** platforms, **banks**, and **telecommunications** companies. When a customer starts a conversation, the bot remembers previous interactions and suggests solutions accordingly.

For example, if a customer recently asked about their order status, the bot might immediately suggest, "Do you want to check your current order status?" when they return.

PRESCRIBED TEXT BOOKS AND REFERENCE BOOKS

TEXT BOOKS:

1. "GANs in Action: Deep Learning with Generative Adversarial Networks" by Jakub Langr and Vladimir Bok
2. "Deep Learning with Python" by François Chollet
3. "Deep Learning with TensorFlow and Keras" by Amita Kapoor

REFERENCES:

1.Goodfellow, Ian, et al. "Generative Adversarial Nets", 2014.

2.Karras, Tero, et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation", 2018.

3.Isola, Phillip, et al. "Image-to-Image Translation with Conditional Adversarial Networks", 2017.

MINI PROJECT SUGGESTIONS

- ✓ Team 1: Develop a Basic In-Context Learning Model for Text Prediction
- ✓ Team 2: Create a Contextual AI Chatbot with Adaptive Response
- ✓ Team 3: Explore Advanced In-Context Learning Techniques in Language Models
- ✓ Team 4: Develop a GAN Model for Image Prompting and Generation
- ✓ Team 5: Investigate Prompt Hijacking in NLP Models
- ✓ Team 6: Address Challenges in In-Context Learning with Continual Learning

THANK YOU

Disclaimer:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.