

Introduction to Prompt Engineering

Generate 



Welcome to Prompt Engineering!

You have likely encountered Artificial Intelligence (AI) in your daily life, whether you realize it or not. This interaction could be through a customer support chat, a suggestion from an AI prompt in the software you use, or a conversation directly with AI tools like ChatGPT or Copilot.

The prompts you input into an AI tool determine the response you receive. To ensure you extract maximum value from AI interactions, you must devise your prompts carefully. This is what Prompt Engineering is all about!

After completing this course, you will be able to:

- Explain the fundamental concepts of Natural Language Processing (NLP) and Large Language Models (LLMs) in the context of prompt engineering.
- Discuss techniques for stakeholder requirement gathering to define desired outcomes from prompts.
- Describe basic prompt structure and formatting.
- Discuss prompt engineering techniques to optimize AI-generated outputs.
- Identify ethical and legal considerations and integrate them as constraints in prompt development.
- Explain the importance of effective communication and collaboration in prompt engineering.

 **Help and Resources****MODULE 1: FUNDAMENTALS OF AI AND NLP FOR PROMPT ENGINEERING**

 **The Foundation of Prompts** **Introduction to AI Architectures** **Unlocking AI Potential with Prompts** **Knowledge Check****MODULE 2: REQUIREMENTS GATHERING AND ANALYSIS FOR PROMPT DESIGN**

 **Requirements Gathering for Prompt Design** **Knowledge Check****MODULE 3: PROMPT DESIGN**

 **Introduction to Prompt Structure and Formatting** **Mastering the Art of Prompting** **Prompt Engineering Techniques** **Professional Practices for Prompt Engineers** **Knowledge Check****MODULE 4: ETHICS AND COMPLIANCE CONSIDERATIONS IN PROMPT ENGINEERING**

 **Legal and Ethical Compliance in Prompts** **Knowledge Check****MODULE 5: EFFECTIVE COMMUNICATION AND COLLABORATION FOR PROMPT ENGINEERING**

 **Prompt Template Documentation** **Prompt Template Documentation Format and Tools**

 **Communication and Collaboration in Prompt Engineering**

 **Knowledge Check**

 **Conclusion**

Help and Resources

Support

Translations may have been conducted using artificial intelligence (AI) or machine translation technology without human review or editing. Formatting issues may occur. If you have any feedback regarding translation quality, please submit a support ticket using the **Help** button.

Resources

- [Additional Notes.pdf](#)
 - [Transformation-program-infographic.pdf](#)
-

Legal

Copyright © Dell Inc.. All Rights Reserved. Dell Technologies, Dell and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

The Foundation of Prompts

Day-by-day Artificial Intelligence (AI) systems are integrating more into your personal and professional life. As your interactions with AI systems grow, it is essential to learn the skills to effectively communicate with AI tools. Hence, learning skills like Prompt Engineering is not only critical, but necessary to make the best of the groundbreaking AI tools available at your disposal.

A good place to start is to understand the foundations that power AI systems. In this module, therefore we discuss:

- Understanding AI and Generative AI concepts.
- The role and evolution of prompts.
- Technologies behind prompts (LLM, NLP, AI architecture).
- Advancement in AI model - Retrieval Augmented Generation (RAG).
- Introduction to prompt engineering.

So, let us get started!

What is Artificial Intelligence?

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and communicate like humans. AI is now an integral part of modern lives, shaping interactions with technology and solutions to everyday problems.

How familiar are you with AI? Try this simple activity below to find out!

Match the definition with the

≡ 1 The simulation of human intelligence in

Artificial

≡ 2 A subset of AI focused on enabling machines

Machine

≡ 3 A computational model inspired by the human brain, used to recognize patterns and make

Neural

≡ 4 Allows machine to understand and generate

Natural language

≡ 5

A type of AI model trained on massive amounts of text data to understand and

Generative



What is Generative AI?



An image created using Generative AI.

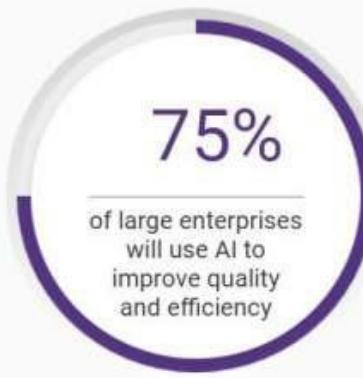
Generative AI (GenAI) is a form of artificial intelligence that creates new content, such as text, images, or music, with minimal human input. Gen AI is conversational and mimics human-like interactions, making it useful for applications such as chatbots and virtual assistants.

The following prompt was used to generate the image on the left: "Create an image of a colorful world with clouds on one side and stars and planets on the other."

Creating this image, which once took hours, now only takes a few minutes with the right prompt!

Is the Hype about Generative AI Real?

The numbers below are an indication on how rapidly AI is integrated into workspaces. The key question is: **Are you prepared to leverage the opportunities in an AI-enabled work environment?**





What are Prompts in Generative AI?

Prompts are essentially the way that you communicate with an AI model in natural language. Like human interaction, the way you phrase your question and the context you provide, influences the response you receive.

Genesis of Prompts

As AI advanced, human interaction with AI tools increased significantly. AI platforms like ChatGPT played a pivotal role in bringing AI into the mainstream, and many workplaces began integrating these tools into their daily operations. This context provided the perfect opportunity to develop the technique of prompt engineering.

However, this evolution did not happen overnight. Let us have a quick look at the evolution of Artificial Intelligence and the prominence of engineering prompts.



Have you Tried Prompts?

If you have ever asked an AI model a question, even something as simple as "What is the weather today?", it means you have used prompts.

Now, let us look at few examples of prompts.



Caution: It is important to keep in mind that AI-generated content can be inaccurate, misleading, entirely fabricated, biased, or offensive. Be sure to carefully review any work containing generative AI content before you use or publish it. Before using a generative AI tool for work, review your organization's guidelines.

Step 1

Helping People with Disabilities Navigate Public Spaces

 Copilot	<p>"Invent a device that helps people with disabilities navigate public spaces more easily. What features would it have?"</p>	Prompt
<p>Inventing a device to help people with disabilities navigate public spaces can be both impactful and innovative. Here are some features such a device could have:</p> <ol style="list-style-type: none">1. Multi-Sensory Guidance:<ul style="list-style-type: none">• Voice Assistance: Provides real-time audio directions and alerts about obstacles or changes in the environment.• Haptic Feedback: Uses vibrations to signal turns, stops, or hazards, which can be especially useful for individuals with visual impairments.2. Advanced Mapping and Localization:<ul style="list-style-type: none">• GPS Integration: Offers precise location tracking and route planning.• Indoor Navigation: Utilizes beacons or Wi-Fi triangulation to provide accurate navigation inside buildings.		Response

In this example, a prompt asks AI to describe the features of a device that helps people with disabilities navigate public spaces. The response describes multi-sensory guidance, such as voice assistance and haptic feedback, and advanced mapping and localization, such as GPS integration and indoor navigation.

Step 2

Innovating Ways to Reduce Air Pollution

Copilot	Prompt	Response
	<p>"Propose a plan to reduce air pollution in urban areas. What innovative technologies or policies would you implement?"</p>	
	<p>Reducing air pollution in urban areas requires a combination of innovative technologies and effective policies. Here's a comprehensive plan:</p> <p>Innovative Technologies</p> <ol style="list-style-type: none">1. Smart Air Quality Monitoring Systems:<ul style="list-style-type: none">• Real-Time Data: Deploy sensors throughout the city to monitor air quality in real-time. This data can help identify pollution hotspots and inform immediate actions 1.• Public Access: Provide citizens with access to air quality data through apps, enabling them to make informed decisions about outdoor activities 1.2. Electrostatic Precipitators with Nanotechnology:<ul style="list-style-type: none">• Enhanced Particle Capture: Use nanomaterials to improve the efficiency of electrostatic precipitators, capturing finer particulate matter (PM2.5) from industrial emissions 1.	

In this example, a prompt asks AI for technologies that reduce air pollution in an urban environment. The response includes a description of Smart Air Quality Monitoring Systems using real time data and Electrostatic Precipitators with Nanotechnology.

Step 3

Developing a Social Media Platform for Mental Health and Well-Being

<p>Develop a new social media platform that prioritizes mental health and well-being. What features does it have, and how does it differ from existing platforms?</p>	Prompt
<p> Copilot Creating a new social media platform that prioritizes mental health and well-being involves integrating features that promote positive interactions, reduce stress, and provide support for users. Here are some key features and differentiators for such a platform:</p> <p>Key Features</p> <ol style="list-style-type: none">1. Wellness Check-Ins:<ul style="list-style-type: none">• Daily Mood Tracking: Users can log their mood daily, with prompts to reflect on their feelings and experiences.• Personalized Insights: Based on mood logs, the platform provides personalized insights and suggestions for improving mental well-being.	Response

In this example, a prompt asks AI to describe features for a social media platform that prioritizes mental health and well-being. The response describes wellness check-ins, daily mood tracking and personalized insights.

Summary

Well-constructed prompts can spark creativity and explore new ideas. These examples offer a variety of prompts designed to inspire creative thinking across different fields, providing valuable resources for writers, artists, designers, and innovators.

Note: The Summary for these examples were written by Copilot and edited by a human for the final result.

What Powers Prompts?

You have seen how AI tools can understand and respond to your questions or prompts in the examples above. But how does AI understand and respond to human language?

The answer lies in **Natural Language Processing (NLP)**. NLP is the technique that allows AI to analyze and generate human-like language, using patterns learned from massive amounts of data.

Join us in this video, as we uncover some of the key NLP techniques that power the prompts.



What are Large Language Models and Generative Pretrained Transformers?

One of the most common models of NLP you probably might have already heard about is the Large Language Model (LLM).

LLMs are a type of AI model trained on massive amounts of text data. They are a powerful tool within NLP, capable of performing tasks like translation, summarization, and question answering.

Generative Pretrained Transformers (GPT) are a specific kind of LLM. They use transformer architecture and are trained in two steps: pretraining on a large

dataset to learn language patterns and fine-tuning on specific tasks to improve performance.

These technologies together power many of today's AI applications—from chatbots to writing assistants—and the video below explains how NLP, LLMs, and GPT work in unison to make it all possible.



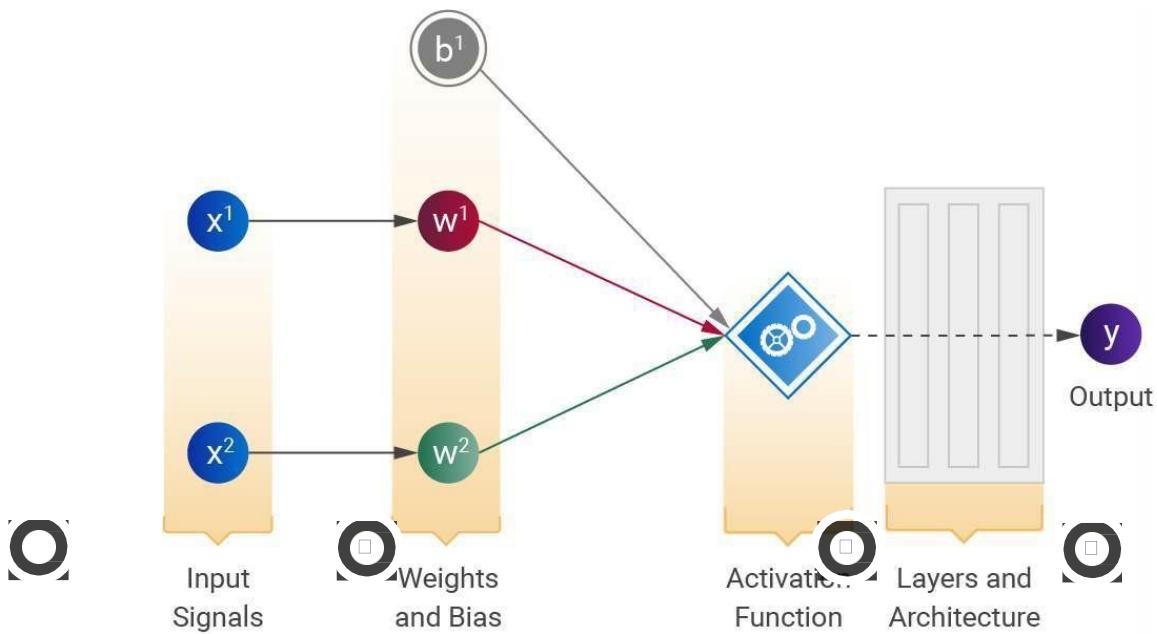
Introduction to AI Architectures

After exploring the basics of NLP and GenAI, it is time to zoom in on the AI architecture that brings these technologies to life. Among the various types of AI architectures—such as decision trees, support vector machines, and rule-based systems—neural networks stand out for their ability to handle complex, unstructured data like language and images.

Neural networks form the backbone of modern NLP and GenAI systems, enabling them to interpret and respond to prompts intelligently. At their core, neural networks are built from interconnected units called **neurons**, which are inspired by the way neurons function in the human brain. This design allows them to learn patterns, make predictions, and generate content in a remarkably human-like manner.

In this module, we will explore the fundamental components and types of neural networks and take a closer look at the strengths and limitations of some AI models.

Components of Neural Networks



Input Signals

Neurons receive input signals from previous layers or directly from the input data.

Weights and Bias

Each input signal is multiplied by a weight. A bias term is added to the sum of these weighted inputs. Weight influences how input data is transformed and learned by the network.

Activation Function

The result is passed through an activation function, which introduces non-linearity and allows the network to learn complex patterns.

Layers and Architecture

Input Layer: Receives raw data.

Hidden Layers: Process data, learning features at various abstraction levels.

Output Layer: Produces the final output based on processed information.

A neural network can be both an AI architecture and an AI model, depending on the context.

Analogy: Think of the neural network architecture as the blueprint of a house, and the neural network model, is a specific instance of architecture, similar to an actual house built using that blueprint, with customized interiors (trained parameters).

Types of Neural Network Architectures

The neurons are organized into different neural network architectures, each suited to specific types of tasks. Common architectures include Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based Neural Networks.

The following table compares various features of key neural network architecture.

Features	Artificial Neural Networks (ANN) Feed-forward Networks	Recurrent Neural Networks (RNN)	Convolution Neural Networks (CNN)	Transformer-based Neural Networks
Primary Use	General-purpose	Sequential data	Grid-like data such as images	Tasks including sequence of text
Architecture	Fully connected layers	Recurrent layers	Convolutional layers	Self-attention mechanisms

Strengths	Simplicity, flexibility	Memory of past inputs	Spatial feature extraction	Parallel processing, context-aware
Example Applications	Classification, regression	Language modeling, time series	Image recognition, video analysis	Machine translation, text generation
Working Mechanism	Consist of an input layer, one or more hidden layers, and an output layer.	Maintain a memory of previous inputs, using feedback loops.	Use convolutional layers to detect features and pooling layers to reduce data dimensionality.	Use self-attention mechanism, which considers the entire context of a sentence, when generating output.

Among these architectures, transformer-based models provide a breakthrough for GenAI. Their self-attention mechanism enables AI to understand and generate contextually relevant content, significantly enhancing output quality.

Self-Attention in Action: "What is the capital of France?", when the model reads this question, it uses self-attention to figure out which words are most important.

- It sees the word "capital" and looks for what it relates to.
- It finds "France" and realizes that the question is asking about the capital of France.
- So, it focuses more on "France" to help generate the correct response: "The capital of France is Paris."

Self-attention helps the model understand how words in a sentence are connected, even if they are far apart.

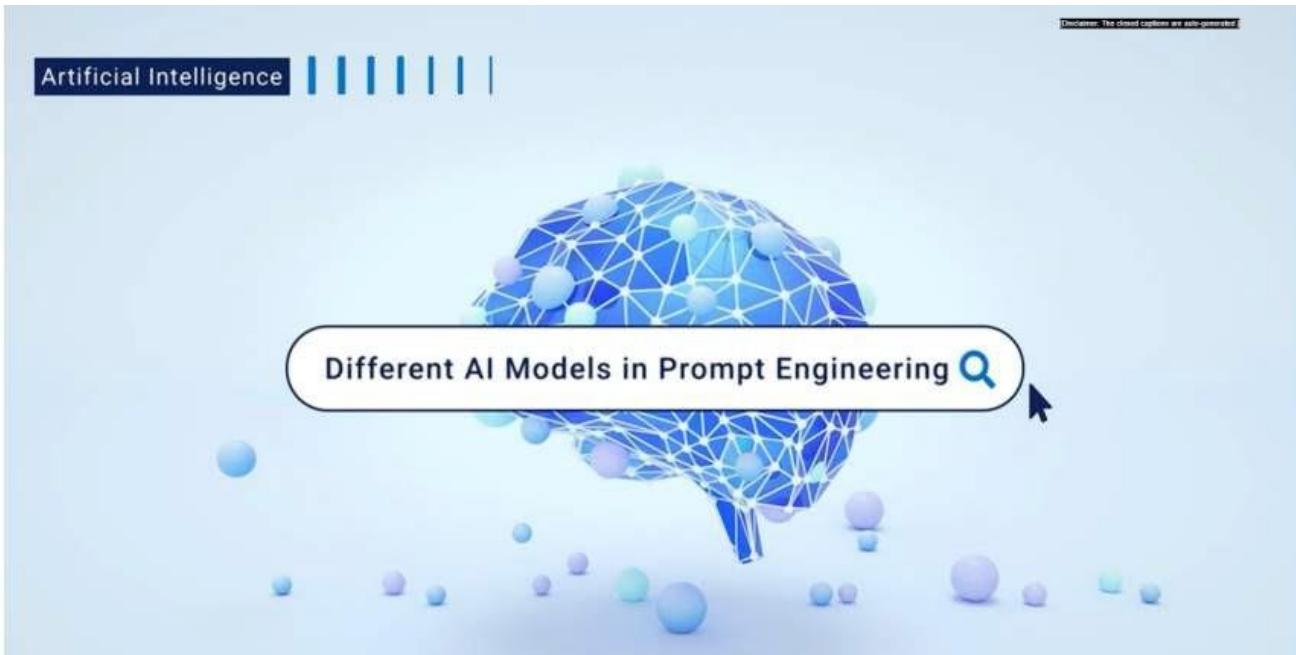
Neural network architecture can be trained in different ways depending on the application. Models can be pretrained on **domain-specific** data (e.g., legal or

medical texts), developed as **foundational models** trained on vast and diverse datasets, or fine-tuned into **custom models** tailored to specific organizational needs.

Understanding the Strengths and Limitations of AI Models

It is important to keep in mind that AI models have limitations, and these limitations impact how models respond to user prompts.

For example, you may not get a solution to an engineering problem from a model designed for medical practitioner use. In this video, we compare and contrast various AI models:

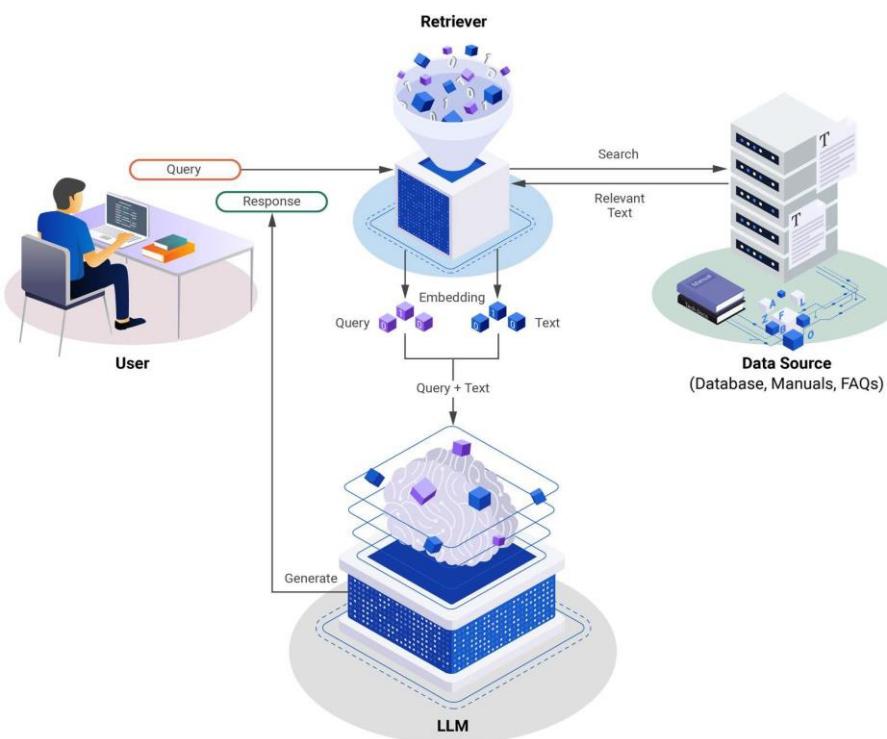


AI models depend exclusively on trained data for generating knowledge, which can sometimes be a limitation. To overcome this limitation, there is an advanced AI model that is called Retrieval Augmented Generation (RAG) can enhance the knowledge of LLMs.

Retrieval Augmented Generation Technique

RAG model combines the power of retrieval-based systems with existing generative AI models to produce more accurate, informed, and context-aware responses.

Integrating a retrieval mechanism allows the model to pull relevant information from external sources in real-time, making it especially powerful for tasks that require up-to-date or specialized knowledge.



RAG Technique in action

RAG technique typically has two parts: a Retriever and a Generator.

- The `retriever` searches and finds relevant information that is based on a given input or prompt.
- The retrieved documents are then combined as context with the original input prompt and fed to the LLM text `generator`. By leveraging both the input and the retrieved information, the text generator generates relevant and coherent responses.



RAG can also be considered as Prompt Engineering technique and examples are covered with prompt engineering techniques later in the course.

Refresh your memory by matching the right AI architecture with its description.

Match the following.

1	Transformer	Specialized in processing grid-like data such as images; extracts spatial features.
2	Convolution Neural Network	Designed to handle sequential data by maintaining memory of past inputs.
3	Recurrent Neural Network	Uses self-attention mechanisms for parallel processing of sequences; backbone of models like GPT.

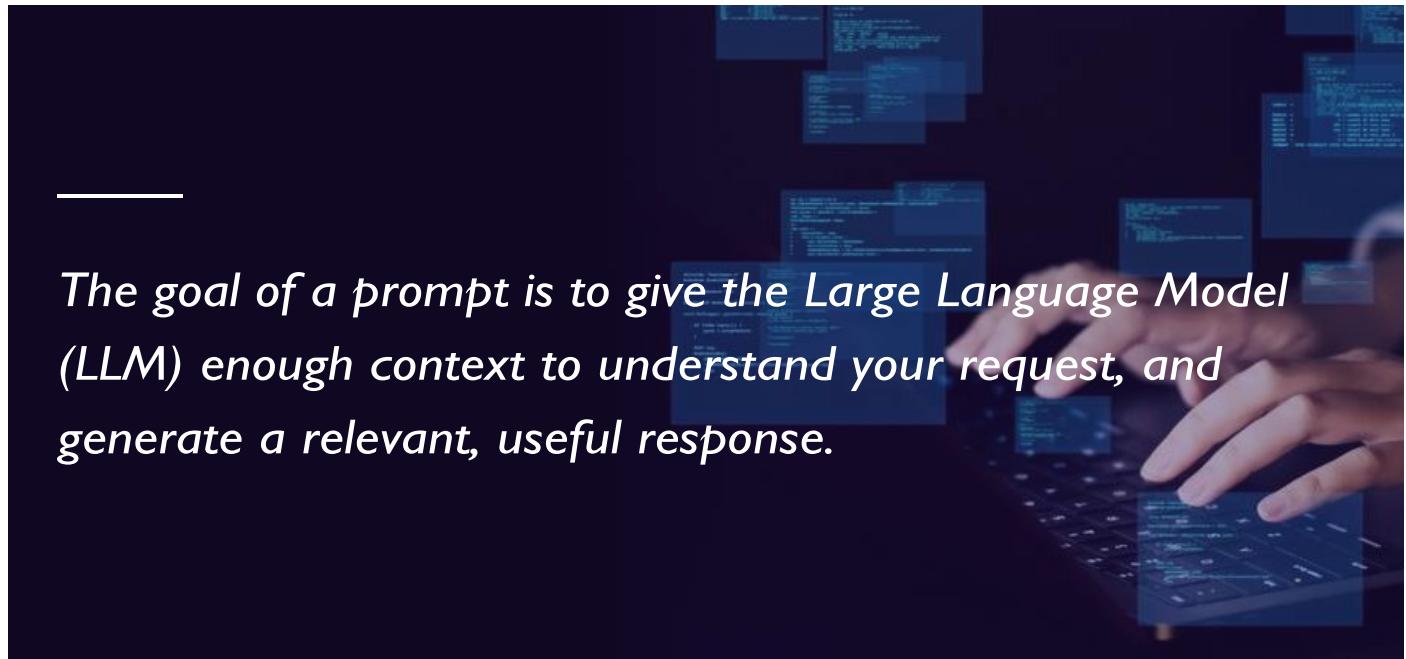
Unlocking AI Potential with Prompts

By now, you must have realized that AI models are complex, with diverse underlying architectures. How AI models are built and used can vary depending on their intended purpose.

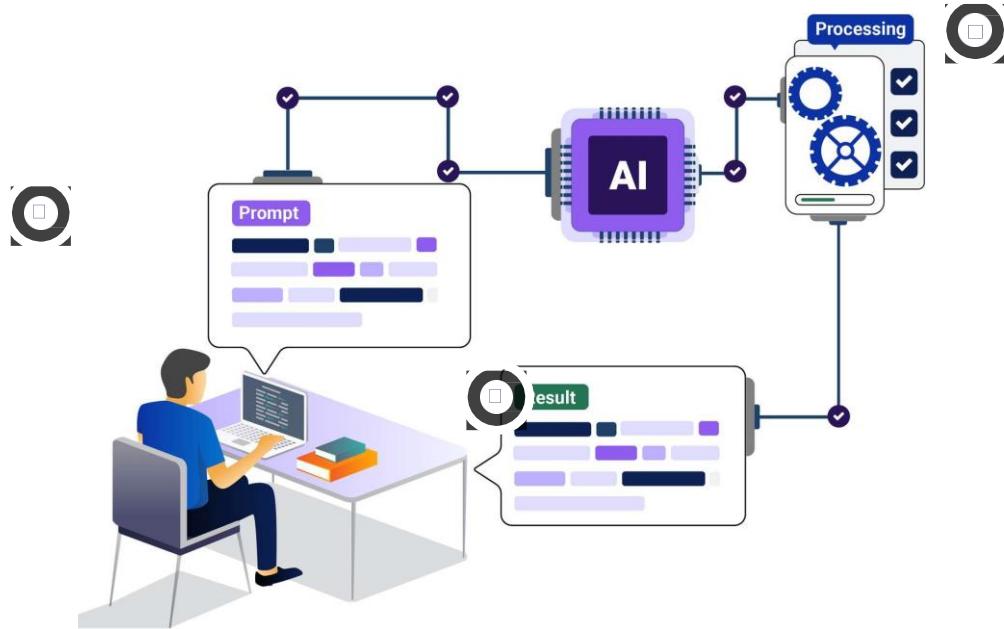
The next step is to understand how you interact with these models using prompts. The data, phrases, or inquiries you input into a generative AI tool are commonly coupled with prompts.

Fun activity: Think of a famous fictional AI from a book or movie. Use an AI tool of your choice to ask, what AI model it might be based on?

The goal of a prompt is to give the Large Language Model (LLM) enough context to understand your request, and generate a relevant, useful response.



Below is the sequence of interacting with AI using prompts.



Prompt

The prompts you provide to a generative AI tool significantly impact the quality of the responses you receive.

Processing

Once you enter a prompt, the AI model processes your input and generates a reply based on its learned patterns.

Results

The AI model produces a response. Providing more detailed prompts can enhance the quality of the output.

The more specific the prompt is, the better the result will be.

- **Basic Prompt:** "Find me a flight from Kansas City to Austin"
- **Refined Prompt:** "Find me direct flights from Kansas City to Austin with the shortest duration and lowest cost on United Airlines"

What is Prompt Engineering?

Now that we know what prompts are, the next step is to understand Prompt Engineering. Prompt Engineering is a learned skill that asks the right questions to get the appropriate output from an AI tool.



A User Trying Prompts

Prompt Engineering:

- Helps you get better results from large language models (LLMs).
- Allows you to guide the model's behavior more precisely.
- Plays a key role in building real-world, useful AI applications.
- Bridges the gap between what you want and what the AI understands.



Understand how NLP concepts and AI models influence prompt interpretation.

Scenario: You are designing a prompt for a Customer Support chatbot for a healthcare platform. The chatbot must assist users in scheduling appointments, providing medication reminders, and answering basic health-related FAQs.

Based on this scenario, reflect on what you have learned so far. Open the first tab below for the reflection question. When you are ready for the answer, open the second tab.

Reflection Question

You enter "Check when my next appointment is" as the initial prompt for the chatbot. Which AI architecture (e.g., CNN, RNN, Transformer) would best support this prompt and why?

Answer

The **Transformer architecture** would best support this prompt. Transformers, such as those used in GPT models, handle contextual understanding more effectively than RNNs or CNNs. They allow for better handling of ambiguous phrases like “my appointment” by understanding conversational context, user session data, and previous dialogue history—making them ideal for dynamic, real-time chatbot

interactions. Because the chatbot has been trained on medical settings it has the context to respond to the prompt.

Summary

We now have a basic understanding of how AI models work and the ways we can interact with them using prompts.

This understanding is key to helping us build effective prompts, and this process is called Prompt Engineering.

Knowledge Check

Let us recap our understanding of AI Fundamentals with a quick knowledge check.

You are working on a chatbot project for a Customer Support team. The chatbot must understand customer queries and respond meaningfully. Which of the following technology is best suited for enabling this capability?

- Prompt Engineering
- Natural Language Processing
- Robotic Process Automation
- Machine Learning

SUBMIT

What is the primary advantage of the Retrieval Augmented Generation (RAG) technique?

- Uses only pretrained knowledge to generate the desired output.
- Retrieves and incorporates external knowledge during generation.
- Uses rule-based logic instead of machine learning to retrieve data.
- Eliminates the need for training the Generative AI model.

SUBMIT

You are asked to improve the responses that are generated by a virtual assistant who is powered by a large language model. The responses are often vague or off-topic. A senior developer suggests, "Try rephrasing your inputs or giving clearer instructions to get better results from the model." What practice is the developer referring to?

- API tuning



Data Labeling



Model retraining



Prompt Engineering

SUBMIT

Requirements Gathering for Prompt Design

Not all AI models are the same. They can be tailored to serve specific needs. For example, a model that is designed for healthcare may be trained to assist medical staff with clinical terminology and patient interactions. Understanding who uses the model and what they need is critical to building an effective and reliable solution.

This module explores how to identify user needs, often referred to as stakeholder requirements, to ensure that AI models deliver meaningful value.

- “ When using generative AI models like ChatGPT or Copilot casually, experimenting with prompts is cost-free. However, in professional settings with specialized AI models, each interaction can be costly and resource intensive. Therefore, it is crucial to design prompts thoughtfully and strategically. Gathering clear requirements from stakeholders ensures that prompts are precise, efficient, and aligned with business needs, maximizing the value of the AI model.

Stages of Requirements Gathering

Designing effective prompts starts with a structured approach to requirements gathering. This process includes:

1. Understanding stakeholder needs.
2. Identifying the target audience.
3. Defining success metrics.
4. Setting clear objectives.

Select the tabs below to navigate the stages of requirements gathering.

GATHER AND ANALYZE STAKEHOLDER REQUIREMENTS	IDENTIFY TARGET AUDIENCE AND SUCCESS METRICS	DEFINE CLEAR OBJECTIVES FOR PROMPT DEVELOPMENT
<p>Identify Key Stakeholders: Users, developers, and business owners are other crucial team members.</p> <p>Conduct Interviews and Surveys: Understand stakeholder needs and expectations.</p> <p>Document Requirements: Comprehensive documentation that uses clear and concise language.</p>		

GATHER AND ANALYZE STAKEHOLDER REQUIREMENTS	IDENTIFY TARGET AUDIENCE AND SUCCESS METRICS	DEFINE CLEAR OBJECTIVES FOR PROMPT DEVELOPMENT
<p>Identify the Target Audience: Their demographics, needs, and preferences.</p> <p>Define Success Metrics: How to measure the effectiveness of the AI application.</p> <p>Analyze User Feedback: How to understand if the generated text is what the end user requires.</p>		

GATHER AND ANALYZE STAKEHOLDER REQUIREMENTS	IDENTIFY TARGET AUDIENCE AND SUCCESS METRICS	DEFINE CLEAR OBJECTIVES FOR PROMPT DEVELOPMENT
<p>Objective Requirements: Translate business goals and user needs into specific prompt requirements.</p> <p>Output Requirements: Define the desired output format style.</p> <p>KPIs: Identify key performance indicators (KPIs) for prompt effectiveness.</p> <p>Development Plan: Create a prompt development plan.</p>		

Let us explore a case study that illustrates how a structured approach to requirements gathering

leads to effective prompt design.

Case study: Devising prompts for enhancing customer support —

An e-commerce company is facing challenges with its customer support system. Despite having a dedicated support team, the company struggles to identify and resolve customer issues. To improve the efficiency of its support process, the company is exploring the use of prompts to streamline operations, reduce response times, and increase overall customer satisfaction.

Below are the steps the development team should take before beginning actual prompt development.



Gathering and Analyzing Stakeholder Requirements



Identifying Target Audience and Success Metrics



Defining Clear Objectives for Prompt Development

Phase 1: Gathering and Analyzing Stakeholder Requirements

Identifying Key Stakeholders

- **Users:** Customers who frequently contact support with order-related queries.
- **Developers:** The software team responsible for designing and implementing the prompts.
- **Business Owners:** Key decision-makers such as the Head of Customer Service and the CTO.

Conducting Interviews and Surveys

- **Interviews:** Conduct one-on-one interviews with a sample group of customers to understand their pain points and expectations.
 - **Example:** Customers may express a desire for faster responses and more accurate information.
- **Surveys:** Distribute surveys to a broader customer base to collect data on common issues and desired features in a support system.
 - Prepare structured questions that align with the project's overall objectives to ensure consistent, actionable feedback.

Documenting Requirements

- **Use Clear Language:** Requirements should be written in simple, concise terms.

- **Example:** “The prompts should provide real-time order tracking and assist with return processing.”
- **Comprehensive Documentation:** Consolidate all findings into a well-organized requirements document that clearly defines the scope, goals, and specific features needed for prompt development.

Phase 2: Identifying Target Audience and Success Metrics

Identifying the Target Audience

- **Demographics:** The target audience consists of tech-savvy millennials who prefer online shopping and expect fast, efficient customer support.
- **Needs and Preferences:** Surveys reveal that customers value quick, accurate responses and personalized interactions.

Defining Success Metrics

- **Response Time:** Reduce the average response time to under 30 seconds.
- **Accuracy:** Achieve a response accuracy rate of at least 95%.
- **Customer Satisfaction:** Aim for a satisfaction score of 4.5 out of 5 in post-interaction surveys.
- **Relevance:** Ensure responses remain on topic and meet at least 90% of user requests.
- **Efficiency:** Optimize the use of computational resources and reduce AI response latency.
- **Consistency:** Ensure that similar prompts produce consistent responses to build user trust.

User Feedback Analysis

- **Collecting Feedback:** Gather customer feedback via surveys and monitor social media channels for unsolicited input.
- **Analyzing Feedback:** Use sentiment analysis to classify responses as positive, negative, or neutral. Apply topic modeling to detect common themes and recurring issues.
- **Improving Prompts:** Use insights from feedback analysis to refine prompts, addressing frequent concerns and enhancing the overall user experience.

Phase 3: Defining Clear Objectives for Prompt Development

—

Translating Business Goals and User Needs into Specific Prompt Requirements

- **Business Goals:** Improve customer satisfaction and reduce support costs.
- **User Needs:** Deliver quick, accurate, and personalized responses.
- **Prompt Requirements:** Develop prompts that handle common queries efficiently and empathetically.

Defining the Desired Output Format and Style

- **Format:** Responses should be clear, concise, and logically structured.
- **Style:** Maintain a polite, empathetic, and professional tone in all interactions.

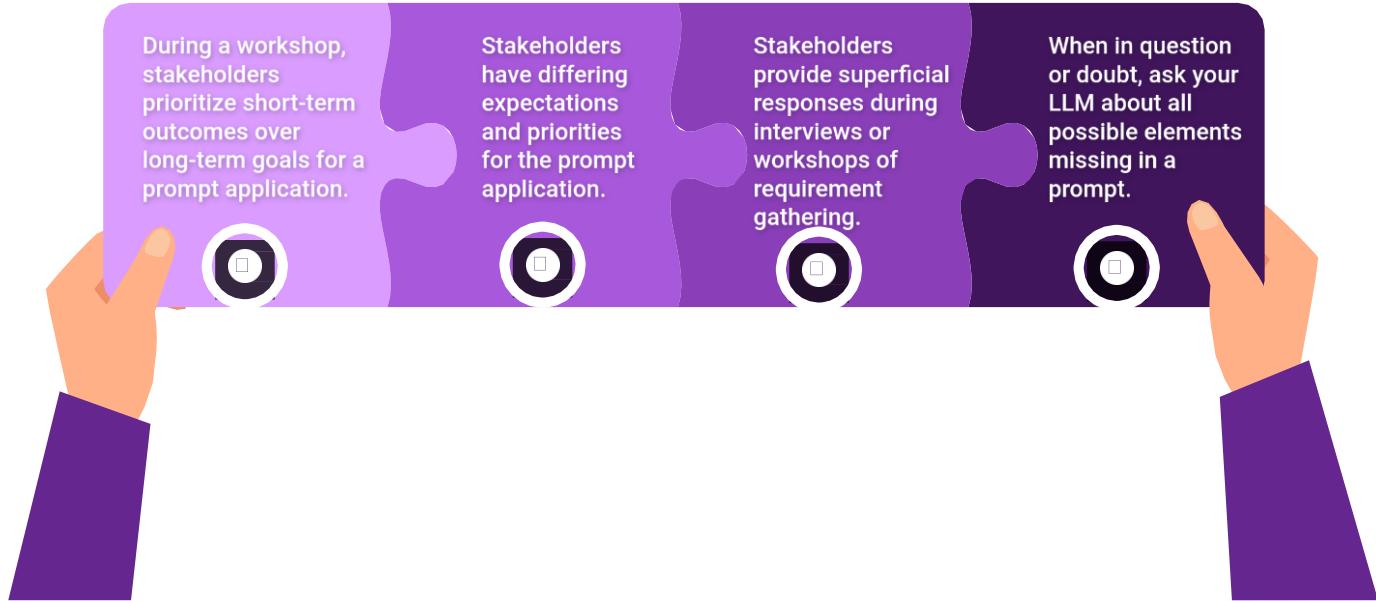
Creating a Prompt Development Plan

Step-by-Step Plan:

1. **Ideation:** Brainstorm initial prompt ideas based on stakeholder input and user requirements.
2. **Design:** Define the structure, context, and flow of each prompt.
3. **Development:** Write and refine prompt content to ensure clarity, accuracy, and alignment with goals.
4. **Testing:** Evaluate prompt performance through internal testing and user feedback.
5. **Optimization:** Adjust and improve prompts based on test outcomes and feedback insights.
6. **Deployment:** Implement prompts within the customer support system or AI interface.
7. **Maintenance:** Continuously monitor prompt performance and make updates as needed to ensure continued effectiveness.

Mitigating Gaps in Requirement Gathering

Select each 'i' icon to mitigate common shortfalls when devising prompts.



Educate Stakeholders

Provide information on the potential risks of neglecting long-term goals and the benefits of a balanced approach.

Facilitate Consensus Building

Conduct clarification sessions to align goals and reach mutual understanding.

Use Probing Questions

Ask targeted follow-up questions to explore underlying needs and gather deeper insights.

Example

"This prompt is not generating the expected results that are [RESULTS], analyze and identify those elements that are missing. This is the prompt [PROMPT]"



Engage with stakeholders to identify key user needs and business goals behind the prompt.

Scenario: You are designing a prompt for a Customer Support chatbot for a healthcare platform. The chatbot must assist users in scheduling appointments, providing medication reminders, and answering basic health-related FAQs.

Using the same scenario, create a prompt plan for the chatbot. Open the first tab below for the reflection questions. When you are ready for the answer, open the second tab.

Reflection Questions

In your plan consider the following:

- User identification - Who are the users of the chatbot?
- User requirements - What tasks do the users need the chatbot to perform?
- Success metrics - How will you measure the success of the chatbot?
- Functional requirements - What functionalities should the chatbot have?
- User experience - What tone and style should the chatbot use to communicate?

Answer —

Prompt Plan for Healthcare Assistant AI Chatbot

- User Identification: Patients and caregivers
- User Requirements: Schedule appointments, provide medication reminders, answer FAQs
- Success Metrics: Appointment scheduling success rate, user feedback scores, response accuracy
- Functional Requirements: Appointment scheduling, medication reminders, FAQ access
- User Experience: Tone- friendly and professional

Summary

Understanding who, where, and why an AI model is used helps design prompts that align with the need, which is the essence of stakeholder requirements. This understanding ensures building prompts that suit the deployed model and extract meaningful value.

Knowledge Check

Let's recap our understanding of requirements gathering with a quick knowledge check.

Scenario: Your team is developing an AI tool for customer service. Due to time constraints, you skip detailed interviews with customer service representatives, assuming their needs are well understood. After deployment, the AI provides irrelevant responses, frustrating customers.

Question: What was the critical mistake made by the team during the requirements gathering phase?

- Conducting thorough interviews with all stakeholders.
- Overlooking the detailed input from customer service representatives.
- Documenting requirements clearly.
- Identifying key stakeholders.

SUBMIT

A company is developing a prompt for a language model to generate personalized product recommendations. What is the first step in defining objectives for prompt development?

- Develop a complex prompt with multiple recommendation algorithms immediately
- Identify the specific customer preferences and expectations
- Focus on recommending every product available in the inventory
- To collect data on competitors' recommender systems

SUBMIT

Introduction to Prompt Structure and Formatting

Having explored the fundamental concepts, technologies, and key considerations of prompt engineering, we are now ready to begin the prompt creation process. The goal is to optimize techniques for getting the best response from the chosen AI model. The aim of using AI is to save time and effort, and crafting the right prompts is crucial to achieving this objective.

A carefully designed prompt can direct an AI model, particularly a Large Language Model (LLM), towards a required outcome. This module explores the fundamentals of prompt creation, formatting, and key techniques.

“ Mastering prompting techniques makes you a better researcher, writer, and decision-maker. The more structured and detailed your prompt, the higher the quality of the generated content.

Get started on the fundamentals of prompt creation.

Every word in a prompt matters. Even subtle changes in phrasing can lead to very different outputs from an AI model.

- "Explain the significance of the **Mona Lisa**" may lead to a discussion on the painting's historical and cultural value.
- "Describe the appearance of the **Mona Lisa**" will likely result in a visual description.

1 of 3

Understanding the subtleties of prompt formatting is also crucial.

AI models, trained on vast datasets, generate responses based on the cues and structure of your input. It's not just about asking a question; **it's about phrasing it in a way that aligns with your intended outcome.**

2 of 3

Ultimately, creating an effective prompt is both an art and a science.

It is an art, because **it requires creativity, intuition, and linguistic finesse**. It is a science, because it draws upon an understanding of how AI models interpret and respond to language.

Components of a Basic Prompt

View the video to explore the components of a basic prompt and some tips for crafting effective prompts.



Examples of Basic Prompts for Various Use Cases

The screenshot shows the Microsoft Copilot interface. At the top is the Copilot logo with the M365 icon. Below it are six cards representing different use cases:

- Get more done**: What are 5 - 6 ways to beat procrastination?
- Improve your writing**: Rewrite this to sound more professional and less verbose:
- Explore what's possible**: Tell me an interesting fact and hypothesize what it reveals about the world
- Get instructions**: Give me step-by-step instructions for making an Instagram Reel
- Track changes over time**: Create a line chart showing the relationship between inflation rates and interest rates over the last 20 years.
- Sharpen your writing**: Analyze the text in the following and make suggestions on how to improve it: [text]

A blue callout box with white text is overlaid on the interface, containing the instruction: "Use attach icon at the input box to attach files into prompts". A large blue arrow points from this callout down towards the bottom right corner of the interface, specifically to the "View prompts" button and the file attachment icons.

Ask me anything...
0 / 16000

View prompts

Co-pilot interface: showing attach file option

How to attach a file to your prompt?

Some LLMs, like Copilot, allow files to be attached to prompts, if the input data is too big for the prompt. The input data can be attached as file using the attach option in the input box. The attachment has to be linked to the prompt (like in the [attached file](#)).



Try this **prompt** on the attached document above:

"I am creating a report for my stakeholders. I have to create a top line for my report that gives them the most important information they should know. Summarize this text and identify the three most important takeaways to put in my top line from the attached document."

EXAMPLE: CREATIVE WRITING PROMPT	EXAMPLE: PROMPT FOR SUMMARIZING TEXT	EXAMPLE: PROMPT FOR PROBLEM SOLVING	EXAMPLE: SIMPLE INFORMATIONAL QUERY
Instruction Write a short story. Context The story should involve a dragon befriending a knight. Input Data None required. Output Format A short narrative. Prompt "Write a short story about a dragon who befriends a knight." Expected Response "Once upon a time, in a land far away, a dragon named Drake, and a brave knight named Sir Cedric formed an unlikely friendship"			

EXAMPLE: CREATIVE WRITING PROMPT	EXAMPLE: PROMPT FOR SUMMARIZING TEXT	EXAMPLE: PROMPT FOR PROBLEM SOLVING	EXAMPLE: SIMPLE INFORMATIONAL QUERY

Instruction	Summarize key points.
Context	The article is about climate change.
Input Data	The article on climate change which is to be summarized.
Output Format	A brief summary.
Prompt	"Summarize the key points of the article on climate change."
Expected Response	"The article discusses the impact of climate change on global temperatures, sea levels, and weather patterns. It emphasizes the need for immediate action to reduce carbon emissions."

EXAMPLE: CREATIVE WRITING PROMPT	EXAMPLE: PROMPT FOR SUMMARIZING TEXT	EXAMPLE: PROMPT FOR PROBLEM SOLVING	EXAMPLE: SIMPLE INFORMATIONAL QUERY

Instruction	Solve a math problem.
Context	The problem involves basic algebra.
Input Data	The equation $2x + 3 = 7$.
Output Format	The solution to the equation.

Prompt	"Solve the equation $2x + 3 = 7$ "
Expected Response	"The solution to the equation is $x = 2$ "

EXAMPLE: CREATIVE WRITING PROMPT	EXAMPLE: PROMPT FOR SUMMARIZING TEXT	EXAMPLE: PROMPT FOR PROBLEM SOLVING	EXAMPLE: SIMPLE INFORMATIONAL QUERY
Instruction	Provide the capital of France.		
Context	The user wants to know the capital city of France.		
Input Data	None required.		
Output Format	A single sentence stating the capital city.		
Prompt	"What is the capital of France?"		
Expected Response	"The capital of France is Paris."		



Try using some of the example **prompts** from this lesson on your own. Modify the prompts and see how the output changes. We recommend using Microsoft Copilot for this activity. If Copilot is not available or if your organization uses a different LLM, use the tool that you have access to.

Enhancing Prompts

Step 1

Enhancing Prompts Through Constraints

Constraints are limitations included in a prompt to narrow the scope of the AI's response and ensure it adheres to specific rules. These may include parameters such as word count, tone, required content, formatting expectations or any boundaries that help focus the model's output.

For Example:

- Specifying a word limit promotes brevity.
- Defining a tone (e.g., formal, casual, or neutral) aligns the output with the intended communication style.

By incorporating well-defined constraints, **you increase the likelihood of receiving responses that are accurate, relevant, and aligned with your goals.**

Step 2

Example: Translation Prompt with Constraints

Instruction	Translate the given technical article from English to Spanish.
Context	The article discusses advancements in renewable energy technology.
Input Data	Article that requires translation to be attached.
Output Format	A translated article in Spanish.
Constraints	Translation should be accurate, culturally adapted.
Prompt	"Translate the following text from English to Spanish, ensuring cultural adaptation..."
Expected Response	Translated Spanish text.

Guiding Prompts with Examples

Examples act as reference points that help AI understand the type of response you're expecting. By including sample prompts and their corresponding outputs, you **provide a clear guide for the desired format, style, or level of detail.**

This technique is particularly effective for complex or technical tasks, where consistency and specificity are important. When the model sees an example, it tends to generalize and follow the same structure or logic in its own response.

Use Case: When asking AI to generate a customer support reply, you might first include a sample exchange to illustrate tone, format, and the kind of language to use.

Step 4

Example: Marketing Prompt with Example

Instruction	Write marketing copy.
Context	The product is a new smartphone.
Input Data	None required.
Output Format	A brief marketing message.
Constraints	Copy should be within 40 words.
Example	"Introducing the latest laptop with cutting-edge technology, sleek design, and powerful performance. Perfect for professionals and gamers alike."
Prompt	"Write marketing copy for a new smartphone the output should be a brief marketing message within 40 words. For example, "Introducing the latest laptop with cutting-edge technology, sleek design, and powerful performance. Perfect for professionals and gamers alike."'"
Expected Response (may vary)	"Experience the future with our new smartphone. Featuring a stunning display, advanced camera, and long-lasting battery. Elevate your mobile experience."

Are Constraints and Examples always Necessary?

WHEN CONSTRAINTS ARE NOT REQUIRED

WHEN EXAMPLES ARE OPTIONAL

While constraints can enhance precision, overly strict constraints may limit the outcome.

- In certain scenarios—particularly in creative writing or open-ended exploration, omitting constraints can encourage flexibility and originality. Constraint omission also allows the AI to generate a broader range of ideas and more imaginative responses.
- For example, asking the model to "**Tell a story about space exploration**" without imposing a word limit or tone can lead to unexpected and creative outcomes.

WHEN CONSTRAINTS ARE NOT REQUIRED

WHEN EXAMPLES ARE OPTIONAL

Similarly, examples are not always essential.

- For straightforward tasks or broad, open-ended queries, the AI can often produce satisfactory responses without guidance through examples.
- Simple prompts do not require reference examples to generate clear, accurate responses. For example:
 - "**Define photosynthesis.**"
 - "**What are the benefits of regular exercise?**"

Keeping prompts concise and direct may be more efficient and equally effective.

What makes a good **Prompt**? Select all that apply.

- Clear - Avoids vague or ambiguous wording.
- Specific - Defines exactly what you need.
- Contextual - Provides background information for relevance.
- Lengthy - Uses long and complex sentences.

SUBMIT

Keywords for Creating Prompts

Below are some keywords that can help you build your prompts. Go ahead and try them out.

Keywords

TRANSLATE	RECOMMEND SOLUTIONS
INCLUDE	EXPLAIN CONCEPT
DEFINE	DESCRIBE BENEFITS or EXPLAIN DRAWBACKS
GENERATE IDEAS	SHORTEN, CONCISE, or SUMMARIZE
CREATE A LIST	DESIGN SCRIPT
DETERMINE CAUSE	CREATIVE SURVEY or SURVEY RESPONSES
ASSESS IMPACT	ANALYZE WORKFLOW

Some example prompts

How to summarize the text using a prompt?

Summarize the following content in two lines: {user input}

How to create an email using a prompt?

Write a concise, professional email to my manager Joe informing them that I will be out of the office today due to illness.

How to generate code using prompt?

Generate a python code snippet to calculate the arithmetic series of a given number.

How to create a marketing campaign?

Generate ideas for a marketing campaign to promote Dell PowerEdge servers to enterprise clients.

How to create an employee survey?

Act as an HR manager focused on improving employee experience. Create a survey to gather feedback on compensation, company policies, and workplace culture.

Guidelines for Writing Prompts

Common Pitfalls to Avoid When Writing Prompts

- **Ambiguity:** Ensure your prompt is clear and specific. Avoid vague language that can lead to misunderstandings.

- **Weak Prompt:** "Write about a book."

Issue: This prompt is too vague. The respondent might not know whether to summarize the plot, discuss themes, or give a personal opinion.

Improved Prompt: "Describe the main themes of your favorite book and explain why they resonate with you."

- **Overloading:** Avoid packing too many instructions or questions into one prompt. Break complex tasks into manageable parts.
 - **Weak Prompt:** "Describe your favorite book, explain its themes, discuss the author's background, and compare it to another book you've read."
Issue: This prompt is overwhelming and contains too many tasks at once.
Improved Prompt: "Describe your favorite book and explain its main themes. In a separate response, discuss the author's background."
- **Bias:** Be mindful of language that might introduce bias or assumptions.
 - **Weak Prompt:** "Why do you think the protagonist's actions were justified?"
Issue: This prompt assumes the protagonist's actions were justified, which might not reflect the respondent's view.
Improved Prompt: "Discuss your perspective on the protagonist's actions. Do you think they were justified? Why or why not?"
- **Complexity:** Keep in mind the audience that will receive the output. Match the complexity of the prompt to the audience's level of understanding.
 - **Weak Prompt:** "Analyze the socio-economic factors influencing the protagonist's decisions in the context of 19th-century European literature."
Issue: This prompt might be too advanced for learners unfamiliar with the subject.
Improved Prompt: "Discuss how the protagonist's decisions are influenced by their social and economic background."
- **Lack of Context:** Provide enough background so the respondent understands what you are asking.
 - **Weak Prompt:** "Explain the significance of the main character's journey."
Issue: Without context, it's unclear which journey or character is being referred to.
Improved Prompt: "In the novel 'Treasure Island', by Robert Louis Stevenson, explain the significance of Jim Hawkin's journey to find the treasure."

Writing Prompts of Varying Complexity

Simple Prompts	<p>Prompt: "Describe your favorite book."</p> <p>Purpose: Elicit a personal and straightforward response based on individual preferences.</p>
Moderately Complex Prompts	<p>Prompt: "Compare and contrast the themes of two novels you have read."</p> <p>Purpose: Encourage analytical thinking and draw connections between texts.</p>
Highly Complex Prompts	<p>Prompt: "Analyze the impact of historical events on the development of modern literature, citing specific examples."</p> <p>Purpose: Engage in higher-level critical thinking, synthesis of ideas and subject-specific knowledge.</p>

Writing Prompts for Different Audiences

For small business owners	Write a launch email for a new accounting software targeting small businesses. Keep the email under 200 words. Use a casual yet professional tone. Format the email in bullet points and include a clear call-to-action (CTA) at the end.
For college students	Write a launch announcement for a new study app targeting college students. Keep the message under 200 words. Use a casual yet professional tone. Output Format: Create a social media post that introduces the app and encourages downloads.

Why This Matters?

These prompts highlight how audience, tone, and output format should be considered when crafting prompts. Even when the core objective is the same (launching a product), the delivery and language must be adapted to suit the target audience's expectations and communication channels.

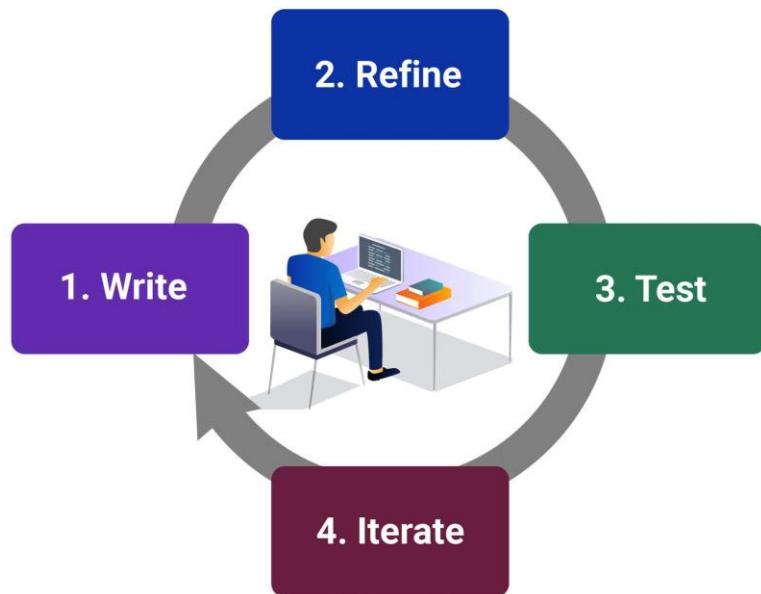


Try using some of the example [prompts](#) from this lesson on your own. Modify the prompts and see how the output changes. We recommend using Microsoft Copilot for this activity. If Copilot is not available or if your organization uses a different LLM, use the tool that you have access to.

Prompt Development Life Cycle

Prompt engineering is a skill, and all skills need practice. The prompt development life cycle is a methodology that provides you with a method to practice and build the perfect prompts.

Prompts cycle: Write > Refine > Test > Iterate



To craft high-quality prompts, follow this four-step framework: Write, Refine, Test, and Iterate. This cycle helps you progressively improve your prompts to consistently elicit better responses from language models.

Step 2

Write the initial prompt

Start by clearly defining the objective, context, and output expectations. Use structured instructions to guide the model.

- **Define the Objective Clearly:** State it as instructions

What is the goal of your prompt?

Example: "Generate an email template for a product launch."

- **Add Context:** Provide background details to improve relevance.

Example: "Write a launch email for a new AI writing tool targeting freelance writers."

- **Provide Input Data:** Specify if any user input or data is required.

Example: None needed.

- **Specify Constraints:** Set word limits, tone, and formatting preferences.

Example: "Keep the email under 200 words. Use a casual yet professional tone."

- **Request a Specific Output Format:** Define the format: list, table, paragraph, Q&A, etc.

Example: "Write in bullet points with a CTA at the end."

Refine the prompt

If the initial output isn't satisfactory, adjust the prompt based on the issues observed.

Key refinements to consider:

- Add constraints like word count, tone or format.
- Include examples to illustrate the desired outcome.
- Clarify ambiguous terms for vague language.
- Specify the level of detail or depth required.

Refined Prompt Example: "Craft a launch email for an exciting new AI writing tool designed for freelance writers. Keep the email under 200 words. Use a friendly yet professional tone. Output Format: Bullet points with a compelling call-to-action (CTA) at the end. Add a personal touch: Include a brief anecdote or personal message to connect with the reader."

Step 4

Test the prompt on LLM and generate a response

Run the prompt through the model and evaluate the output. Evaluate against the following key criteria:

- **Accuracy:** Verify factual correctness; include fact-checking instructions if necessary.
- **Relevance:** Ensure the response aligns with the goals in the prompt.
- **Response Time:** Check that the response time is acceptable for the use case.
- **Customer Satisfaction:** Collect stakeholder feedback through surveys or test runs.
- **Efficiency:** Verify the response uses reasonable computational resources.
- **Consistency:** Check if similar prompts produce similar results.

Iterate

Repeat the cycle until the output meets your desired quality or success metrics.

Tips for iteration:

- **Document Changes:** Track prompt versions and related output.
- **Compare Results:** Assess improvements over iterations.
- **Gather Feedback:** Involve stakeholders to validate outcomes.

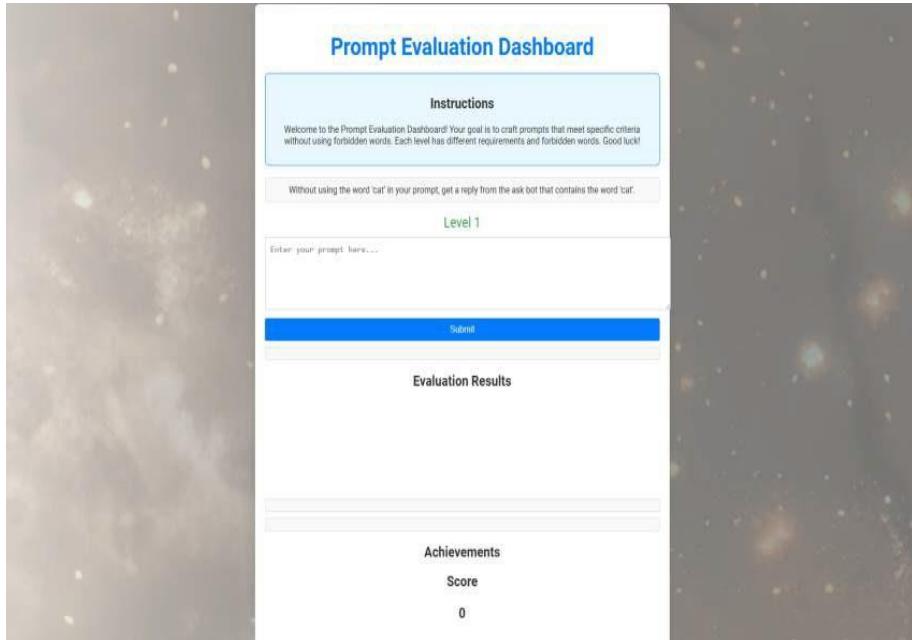
When to stop iterating:

- **Positive Feedback:** Users find the prompt clear, engaging, and helpful.
 - **Desired Outcomes:** The generated responses meet defined goals and success criteria.
- Diminishing Returns:** Further refinements produce minimal or no significant improvement.

Evaluate your Prompt

Let us put your learning into practice. Use the simulation below to test your prompt creation strength.

Refer course for the simulation.



Guess the Correct Prompt.

You want to generate a marketing slogan for a new eco-friendly product. Which prompt do you think yields the best result?

-
- Create a slogan.
 - Create a catchy slogan for a new eco-friendly product that emphasizes sustainability and innovation.

Generate a marketing slogan for a new product that targets luxury consumers.

-

Mastering the Art of Prompting

There is a famous quote that says, "Failure is the stepping-stone to success." While we do not intend to fail, it is critical to understand common mistakes and how to avoid them. Sometimes it is important to know what not to do first! So, let us look at what we must do to avoid your prompts from failing!

When Prompts Miss the Mark



This section highlights potential failures, where the LLM might not fully grasp the request, generate irrelevant answers or give a generic response.

Ambiguous Request



- **User's Prompt:** “Can you help me with this?”
- **LLM's Response:** “Sure, I'd be happy to help! What do you need assistance with?”
- **Issue:** The LLM doesn't know what “this” refers to, so it can't help effectively.
- **Fix:** Be specific. Try: “Can you help me write a cover letter for a software engineering job?”

Cultural or Regional Specifics



- **User's Prompt:** "What's the best local dish in my town?"
- **LLM's Response:** "Local dishes vary widely depending on the region."
- **Issue:** The LLM doesn't know where "your town" is.
- **Fix: Add details. Try:** "What's the best local dish in Austin, Texas?"

Misinterpreted Context



- **User's Prompt:** "Can you tell me about Dell's latest product?"
- **LLM's Response:** "Dell has rebranded its product lines to simplify their offerings."
- **Issue:** The LLM missed the "latest" part and talked about their product line instead.
- **Fix:** Add specificity. Try: "Summarize the new features of the Alienware Area-51 Gaming Desktop."

Off-Topic Responses



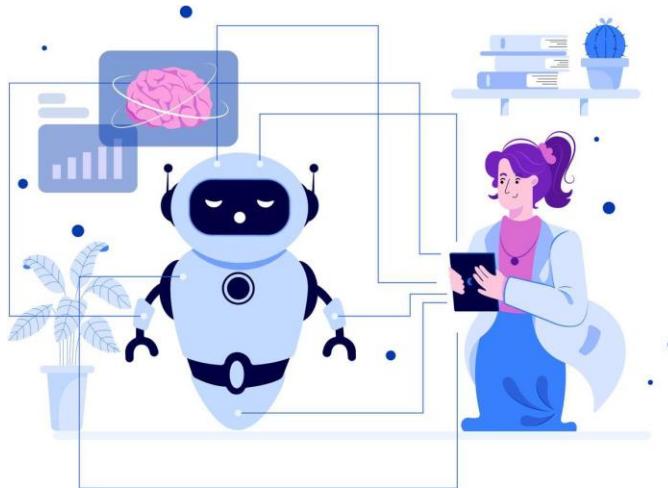
- **User's Prompt:** “What are the symptoms of diabetes?”
- **LLM's Response:** “Eating a balanced diet and exercising regularly are important for maintaining good health.”
- **Issue:** The response is generic and doesn't answer the question.
- **Fix:** Reword clearly. Try: “List the early and common symptoms of Type 2 diabetes.”

Broad Information



- **User's Prompt:** "Tell me about climate change."
- **LLM's Response:** "Climate change refers to long-term changes in temperature and weather patterns."
- **Issue:** The response is technically correct, but painfully generic.
- **Fix:** Focus the prompt. Try: "Explain how climate change is affecting agriculture in South Asia."

Vague Descriptions



- **User's Prompt:** "What is artificial intelligence?"
- **LLM's Response:** "Artificial intelligence is the simulation of human intelligence in machines."
- **Issue:** The response is too basic.
- **Fix:** Ask for more depth. Try: "Give an example of how artificial intelligence is used in healthcare."

Summary



Here's what's usually behind the misfires:

- **Ambiguity:** The LLM doesn't know what "this" means or what you're really asking.
- **Lack of context:** Vague prompts or missing background info can leave the LLM guessing.
- **Overly broad requests:** Broad or open-ended prompts often lead to generic or shallow responses.
- **Too many tasks:** Overloading a prompt can confuse the model or cause it to skip parts.
- **Missing structure:** Without clear formatting cues, the output can become inconsistent or repetitive.

Match the Issues with Solutions

Match each mistake to its corresponding solution.

≡ 1

Lack of Clarity

Ensure your prompt is clear and unambiguous.

≡ 2

Lack of Context

Provide enough context for Copilot to understand the task.

≡ 3

Overly Ambitious

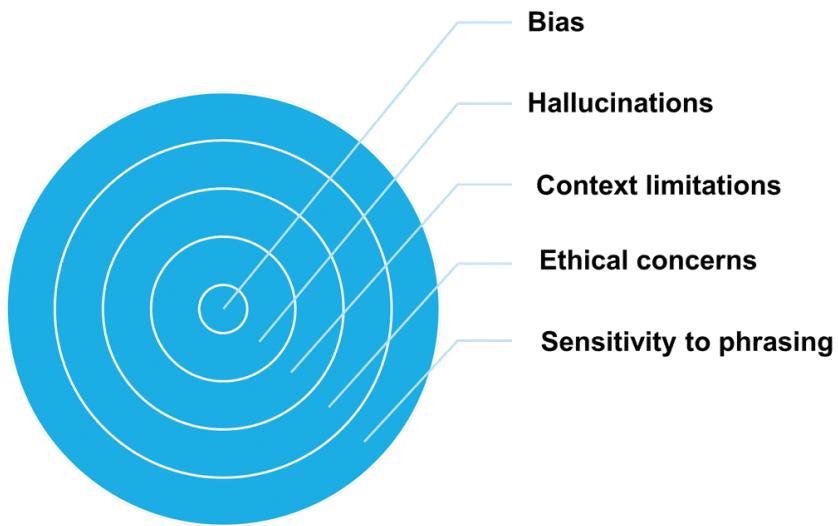
Be realistic about what can be achieved in a single prompt.

SUBMIT

Considerations When Using LLM

While LLMs are powerful, their effectiveness depends greatly on how they are built and fine-tuned. Understanding the limitations of a model or keeping an eye out for potential limitations in the system helps you engineer prompts to mitigate unexpected outputs.

Below is more about each issue.



Bias

Cause: LLMs can reflect biases present in their training data, often mirroring existing societal inequalities.

Detection: Analyze model outputs across diverse datasets to identify disproportionate representations or stereotypical patterns.

Mitigation: Debias the training data, fine-tune the model with fairness constraints, apply adversarial training and auditing methods.

Hallucination

Cause: LLMs may generate incorrect or fabricated information—known as hallucinations—especially when dealing with unfamiliar topics. This happens when the model attempts to fill knowledge gaps with plausible but false responses.

Detection: Compare outputs with verifiable facts or source documents to identify inaccuracies or contradictions.

Mitigation: Use clearly structured fact-based prompts, implement external fact-checking tools and retrieval-augmented generation (RAG) systems and validate responses before use.

Note: Mitigation of hallucination doesn't depend on user input only; it is also the responsibility of the model used.

Though it is not encouraged to test different models, that is the way to overcome it.

Context Limitations

Cause: LLMs can lose track of earlier input in long or multi-turn conversations due to context window limitations.

Detection: Evaluate the model's coherence, consistency, and recall accuracy across extended dialogues or documents.

Mitigation: Use models optimized for long-context handling (e.g., transformer variants with long-range attention), summarize and inject key past interactions into future prompts, and employ session-level memory mechanisms (if supported).

Ethical Concerns

Cause: LLMs can be misused to generate harmful content such as hate speech, misinformation, or malicious code.

Detection: Monitor outputs for toxicity, bias, or unsafe behavior using automated filters and human review.

Mitigation: Implement safety guardrails and content moderation systems, develop and enforce ethical use guidelines, and promote collaboration among developers, researchers, and policymakers.

Sensitivity to Phrasing

Cause: Small changes in wording can lead to dramatically different outputs, making prompt engineering unpredictable.

Detection: A/B test variations of the same prompt and observe the differences in tone, detail, or content.

Mitigation: Use robust prompting techniques, including clear instructions and in-context examples, favor structured inputs over ambiguous or casual phrasing, and leverage prompt templates for repeatable tasks.

Test Your Understanding of Prompt Creation

A data scientist is designing prompts for text classification. What elements should be included in prompts to increase accuracy of identifying diverse text categories.

Include clear examples of each text category to be classified.

A researcher is developing prompts for a language translation model. What should be included in the prompt to ensure accurate translations across different languages?

Provide specific phrases to translate and include any relevant cultural or contextual information.

A content creator is developing an FAQ section for a website. What should be avoided to ensure clarity and usefulness?

Avoid using overly technical jargon that may confuse readers.



Tip: When designing prompts, always consider audience understanding, task clarity, and real-world variability. The more precise and relevant your instructions, the more accurate and usable the AI's response is going to be.

Types of Prompts

Prompts can be classified into different types based on the approach and use case. Let us look at some common types of prompts.

Get to Know System Prompts



This section discusses system prompting and best practices for its use.

What are System Prompts?



Definition: System prompts, particularly with large language models (LLMs), are a set of instructions that define the overall behavior and context for an AI agent or model. It guides how the AI should respond to user prompts, influencing its tone, role, and overall style.

When to Use: System prompts are typically created by developers and used to maintain consistent behavior across sessions. They are rarely visible to end users and are often updated only when necessary.

Key elements of system prompts include:

- **Behavioral Framing:** Define the AI's role, personality, and expertise.
- **Constraint Setting:** Set rules and limitations for tone, format, or types of responses.
- **Context Provision:** Include background information to help the AI perform tasks appropriately.
- **Ethical Guidance:** Embed principles such as fairness, safety, or helpfulness.

System Prompts vs User Prompts



User Prompts:

- Provide turn-by-turn instructions for specific interactions or tasks.
- Dynamic and context-specific, guiding the AI on what to do or respond to in each individual interaction.
- Focused and relevant only for immediate interaction.

Prompt Example:

"Summarize the latest news article about climate change."

System Prompts:

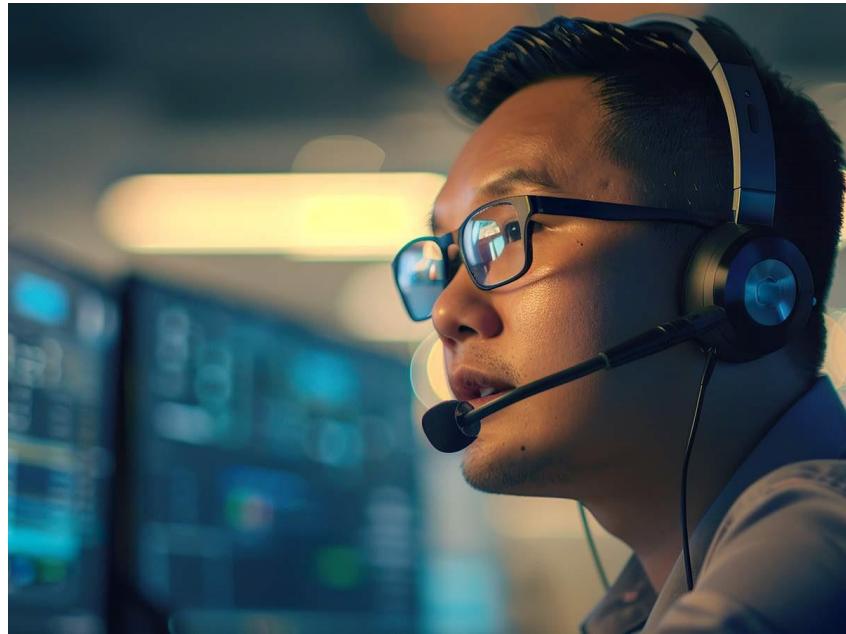
- Set persistent guidelines and foundational instructions for the AI's behavior and responses across all interactions.
- Enable the creation of domain-specific or role-specific AI assistants.

Prompt Example:

"System: You are an experienced AI marketing specialist who specializes in writing persuasive and engaging content for sales communications."

Step 4

Example 1



Prompt:

"System: You are a knowledgeable technical support specialist. Always provide clear and concise assistance. Offer step-by-step guidance for troubleshooting technical problems, ensuring users understand each solution."

Step 5

Example 2



Prompt:

"System: You are a trusted financial advisor. Provide insightful and prudent financial guidance. Analyze financial situations, offer investment advice, and help clients achieve their financial goals."



Tip: System prompts are especially useful in multturn conversations, where the AI must remain consistent in tone, role, or expertise. Getting the initial framing right saves effort in every subsequent exchange.



Try writing system prompts for the following roles:

Prompt: Customer Service Representative (Hint: Consider tone, empathy, and problem resolution)

Prompt: Project Manager (Hint: Think about task delegation, communication, and deadline awareness)

Get to Know Role-based Prompts

- **Definition:** Role-based prompts are designed to instruct an AI model to adopt a specific role or persona that aligns with the task at hand. This helps tailor responses to meet user expectations and maintain consistency throughout the conversation.
- **When to Use:** Use role-based prompts when you want the AI to consistently behave as a specific professional, expert, or character. For example, a teacher, project manager, developer, or a Customer Support agent.

One of the most effective methods for creating role-based prompts is by using an “Act as” clause.



Prompt Example

Act as a social media influencer. You generate content for multiple channels, such as Instagram, Twitter, and YouTube, and interact with followers to raise brand awareness and market products or services. My first idea is, "I need help with creating an engaging Instagram campaign to promote a new line of headsets."



Tip: Role-based prompts improve response quality by aligning the AI's language, tone, and decision-making, with the expectations of that role. This is helpful in marketing, education, technical writing, Customer Support, and more.



[**Try acting as a Business Name Generator using an "Act as" clause.**](#)

Prompt: "Act as a Business Name Generator: I am launching a startup focused on eco-friendly packaging solutions that help reduce plastic waste and promote sustainability. The brand will cater to both businesses and environmentally conscious consumers, offering biodegradable, compostable, and recyclable packaging products. Please suggest a list of catchy, memorable, and brandable business names that reflect our mission of sustainability, innovation, and environmental responsibility."

[**Experiment with other prompts using an Act as clause.**](#)

Combining Role-based and System Prompts

System prompts and role-based prompts can be used together to create highly focused and context-aware AI behavior. This combination ensures consistency in tone and expertise while allowing flexibility in handling specific tasks or scenarios.

1. Establishing Behavior with a System Prompt

The system prompt defines the AI's overall role, tone, and boundaries across all interactions.

"System: You are a Customer Support representative specializing in product returns. Provide clear instructions, empathize with customer concerns, follow company policies, maintain a friendly and professional tone, and resolve issues promptly."

2. Refining Behavior with a Role-Based Prompt

Once the system prompt is set, role-based prompts help guide the AI in responding to specific requests or user inputs.

"Act as a knowledgeable support agent providing detailed instructions for returning a defective product without label."

Why Combine Them?

Using both types of prompts together helps:

- Maintain consistent tone and personality.
- Adapt to specific tasks or edge cases.
- Ensure clarity, helpfulness, and compliance with company or brand guidelines.

Prompt Examples

Technical Documentation

System Prompt

"System: You are an experienced developer. Provide expert guidance, follow best practices, maintain a professional tone, and solve technical issues efficiently. "

Role-Based Prompt

"Act as a mentor explaining complex programming concepts to junior programmers."

Marketing and Content Creation

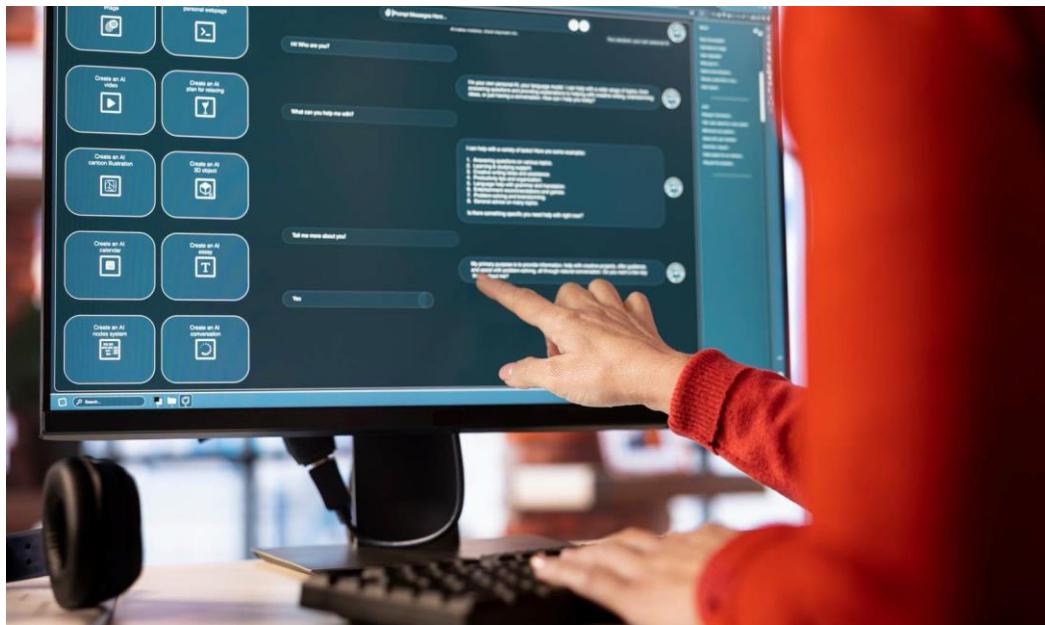
System Prompt

"System: You are a creative copywriter. Craft engaging content, follow brand guidelines, maintain a professional tone, and deliver compelling messages."

Role-Based Prompt

"Act as an enthusiastic startup founder drafting promotional materials."

Get to Know Goal-Oriented Prompting



This section discusses goal-oriented prompting and best practices for its use.

Goal-Oriented Prompts



Definition: Goal-oriented prompting involves clearly stating the desired outcome or objective within your prompt. This helps guide the AI's response to directly support your intention—whether it's solving a problem, making a decision, or generating useful insights.

When to Use: Use goal-oriented prompts when your task has a specific purpose, such as:

- Improving efficiency
- Gaining strategic insights
- Enhancing communication
- Resolving a challenge
- Supporting decision-making

Example



Goal: Improve efficiency in a remote team

Example Prompt: "Suggest methods to streamline workflow in a remote team, focusing on communication tools, task management, and collaboration strategies."

Why it works: The prompt is specific about the objective (efficiency), the context (remote team), and the areas of focus (communication, task management, collaboration).



Tip: Always ask yourself: What do I want to achieve with this prompt? If you can answer that in one sentence, you are ready to turn it into a goal-driven instruction.

Try These Prompts!

- **Troubleshooting a customer network connection:**

Prompt: "[Role] You are a help desk support agent. [Task] Help a customer troubleshoot a problem with their network connection. [Format] Provide the steps in a numbered list."

- **Outlining a new blog post on social media trends:**

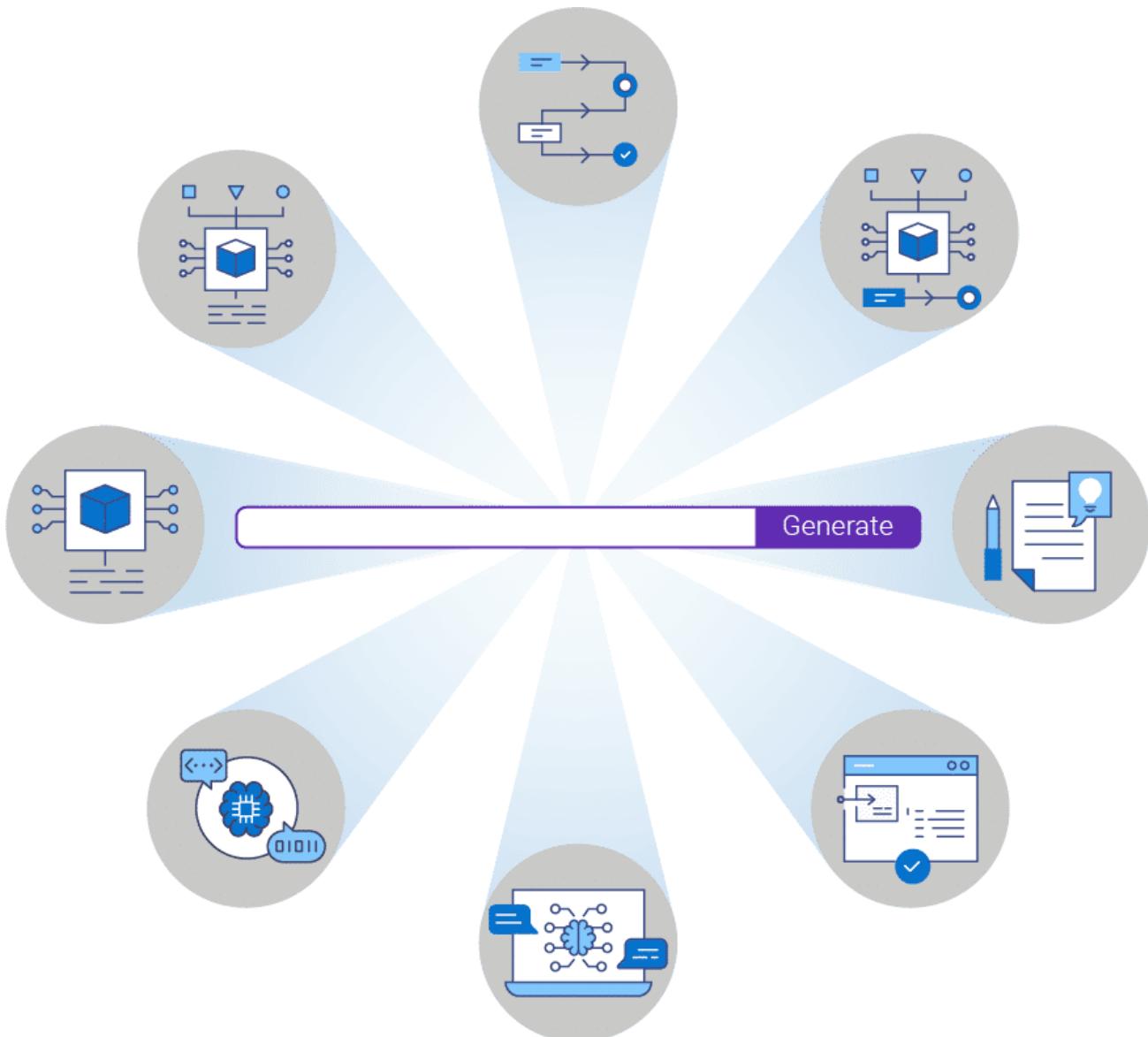
Prompt: "[Role] You are a content writer specializing in digital marketing. [Task] Create a blog post outline unpacking the latest trends in social media marketing [Format] in bullet point form."

- **Prompt for roleplays:**

Prompt: "Imagine you are a medieval knight and a modern-day astronaut. Explain to each other what virtual reality is, using explanations and references that are true to your experiences and knowledge."

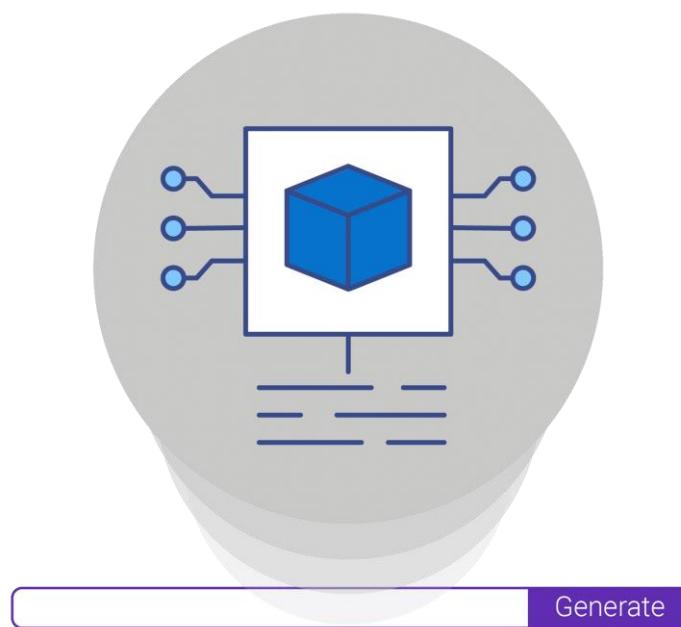
Prompt Engineering Techniques

Prompting Techniques



Based on what you are looking for, the prompt you devise may be relatively simple or a deep philosophical question. Your approach to each use case should be different. Prompting techniques help to identify the best method for your needs. So, let us look at some methods and the needs they address.

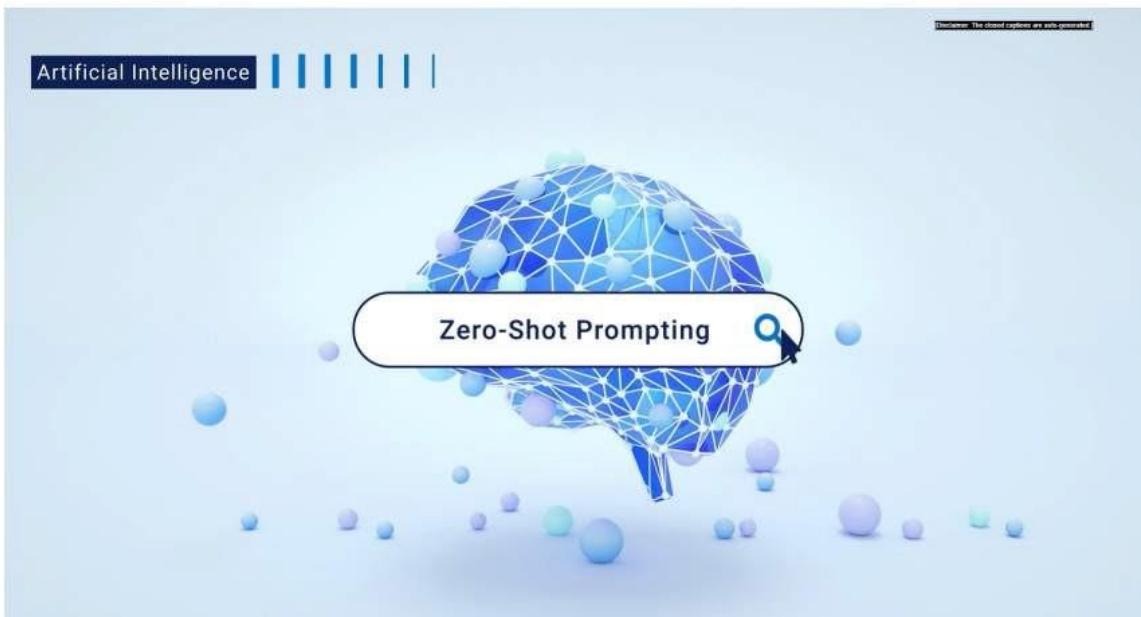
Zero-shot Prompting



Zero-shot prompting involves giving a model a clear, standalone task without supplying any prior examples, training data, or in-context demonstrations. The model relies entirely on its pre-trained knowledge to respond.

Step 2

Examples: Zero-shot Prompting



Example 1: Zero-shot Prompting



A zero-shot prompt would involve providing a prompt like:

- “Generate a response to a customer complaint: ‘I received a damaged product and would like a refund.’”
- “Explain the concept of inflation in simple terms.”

These prompts do not include any guiding examples or reference outputs, only the task itself.

Step 4

Example 2: Marketing and Internal Communication



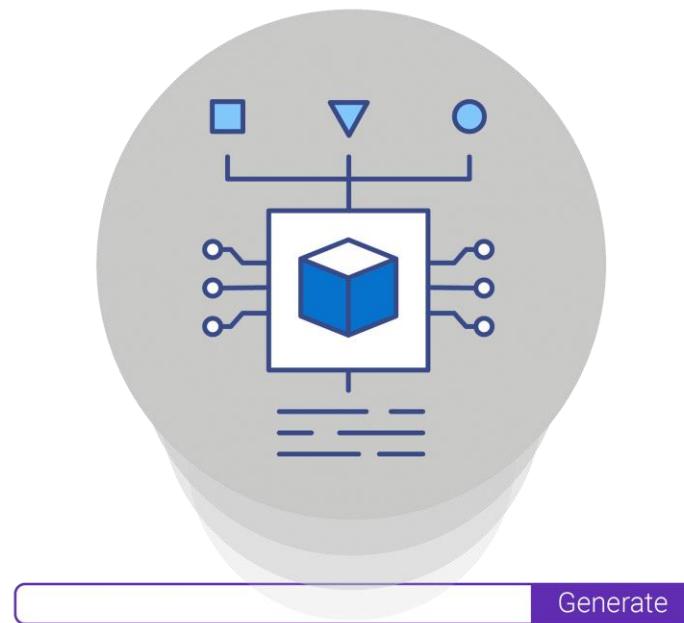
- "Generate a catchy slogan for a new eco-friendly water bottle."
- "Draft an email to inform employees about a new remote work policy."

Each task is framed clearly, yet no additional context or example is given.

Summary

In zero-shot prompting, the model is expected to understand and complete the task based solely on the provided instructions. This method is ideal for simple, familiar tasks or when quick results are needed, without additional setup.

Few-shot Prompting



Generate

Few-shot prompting involves providing a limited number of examples alongside the main prompt to guide the language model's response. This technique helps the model infer the desired style, format, or logic by referencing the patterns in the examples.

Example 1: Few-shot Prompting



In contrast, few-shot prompting provides a few examples or previous data alongside the prompt. For example, we could provide a prompt like,

"Please analyze the sentiment of the following movie reviews: ['The avatar movie was breath taking journey through the realms of fantasy', sentiment: positive], ['The Crow movie was a dull experience and hardly worth the time', sentiment: negative], ['The Dark Night was just okay and delivered what it promised', sentiment: neutral]. Now analyze the sentiment of the following movie review: ['The movie Spider man: No way home was a mesmerizing and engrossing experience']."

Step 3

Example 2: Few-shot Prompting



- "Generate one-line memo statements for an HR collaboration tool. Each memo should be under 20 words and clearly communicate the topic. [Example Topic: Organizational Change, Memo: Effective July 1st, our structure will realign to boost collaboration and streamline decisions]. Now generate a memo for: [Topic: Quarterly Sales Information Update]"

Step 4

Example 3: Few-shot Prompting (List)



"Generate a promotional email that encourages customers to try a new offer. The audience is between 20 to 50 years old. The email should not be more than 30 words. [Offer: Try a new product with a 20% discount. Email: Try Something New – 20% Off Just for You! Discover our latest product and enjoy 20% off your first purchase. It is the perfect time to treat yourself to something fresh. Use code NEW20 at checkout!]. Now write a promotional email for [offer: Flat ₹500 Off on Orders Above ₹2,000] "

Step 5

Example 4: Few-shot Prompting (Summarization)



"Review the below text summarization [Text: The emergence of GenAI and large language models such as Open AI's GPTs indeed presents a range of new risks and regulatory challenges for businesses. Some of the biggest risks of generative AI concern trust and security and include hallucinations, prompt leaking, prompt injection, jailbreaking, privacy, deepfakes and so on. This risk can lead to ethical, operational, compliance to reputational challenges for the organization. Geoffrey Hinton, one of the pioneers in the development of deep learning and neural networks, has stated that "with rapid use of generative AI it has become more important to focus on the dangers and risks of generative AI". Summary: Generative AI like GPTs brings major risks—such as hallucinations, privacy breaches, and deepfakes—posing ethical and reputational challenges, prompting experts to stress the need for greater caution.]. Now summarize text in given

link [\[https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1299.pdf\]](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1299.pdf) "

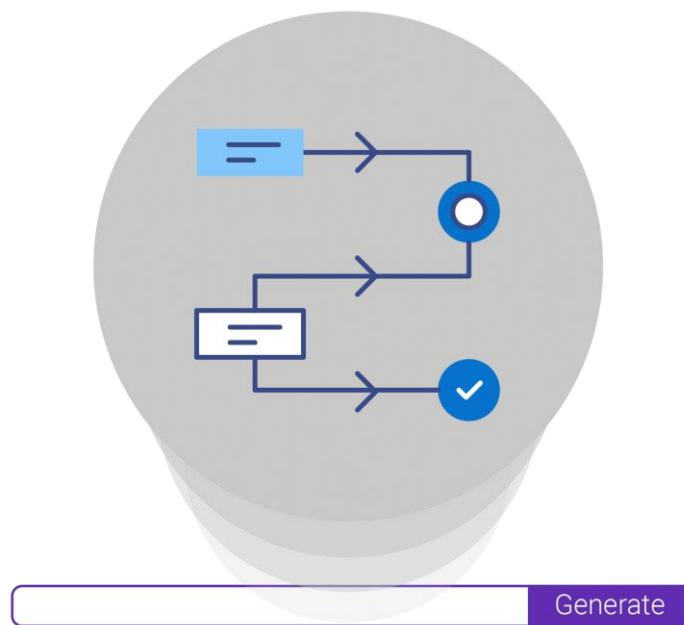
Example 5: Few-shot Prompting (Clauses in Prompts)

"Act as a given role, highlight your key tasks "[**Role:** UI/UX developer, **Key Tasks:** Your task is to enhance a self-service catalog. You will need to provide specific details and devise creative strategies to improve the user experience. This may include designing prototypes, conducting design tests, and offering feedback. Your ultimate goal is to ensure a seamless checkout process for customers]. Now act as Cybersecurity Specialist and define the task.".

Summary

In few-shot prompting, the model generates responses by generalizing from a small set of examples provided in the prompt. This technique is especially effective in scenarios like classification, summarization, and structured content generation where examples can guide consistency and tone.

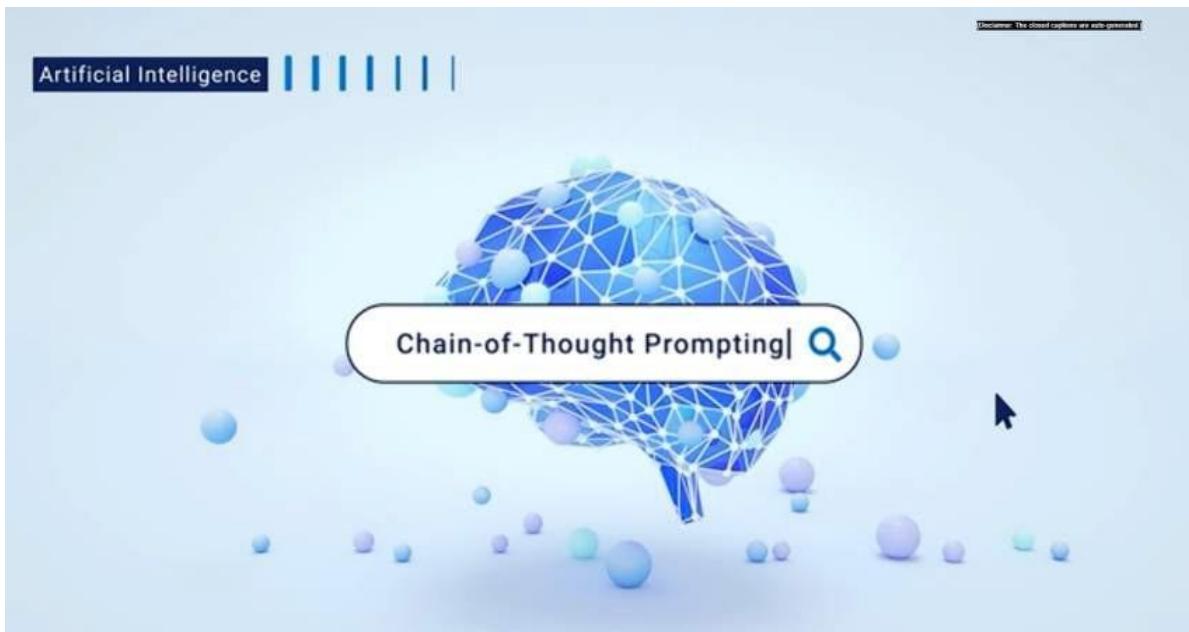
Chain-of-Thought (CoT) Prompting



Chain-of-Thought prompting involves guiding the model to reason through a task step-by-step.

This method helps break down complex tasks into manageable parts, leading to more structured, logical, and accurate outputs. CoT is important to get the most value of reasoning models.

Example 1: Chain-of-Thought Prompting



Let's consider a scenario where we want the model to generate a coherent email on a specific topic. The prompt includes instructions to think through key points before generating the content.

Prompt:

"Think about the following points: [The benefits of adopting sustainable practices in the workplace], [Examples of sustainable initiatives], [How employees can contribute to sustainability efforts]. Now, write an email on the given topic incorporating these points. Think step-by-step."

Why it works: By prompting the model to reflect on each component before writing, the output is more complete, focused, and logically ordered.

Example 2: Chain-of-Thought Prompting



Scenario: A server health monitoring app located on an internal server keeps crashing for users. Your task is to review all related process documents and generate an investigation report based on a user-submitted incident report.

(Note: You can download the 'Sample incident report', below this tab interaction and use it as attachment for this prompt.)

Prompt 1:

"Think step-by-step about how to review all related process documents and generate an investigation report based on the user's incident report about a crashing server health monitoring app."

Tip: Guide the model to cover key areas:

- Investigate system requirements and compatibility
- Check for recent software updates or patches
- Review event logs for error messages

Prompt 2 (With embedded chain-of-Thought):

"Generate a chain of thoughts using the following steps: [Investigate system requirements and compatibility], [Check for recent software updates or patches], [Review event logs for error messages]."

Why it works: Structuring the thought process helps the model prioritize information and produce a clear, actionable report.

Summary

Chain-of-Thought prompting improves output quality by encouraging the model to "think" before generating. This technique is especially valuable for tasks that involve:

- Logical reasoning
- Decision-making
- Troubleshooting
- Multi-step writing tasks

By breaking down the task into a sequence of logical steps, CoT prompts help large language models produce coherent, detailed, and reliable results.

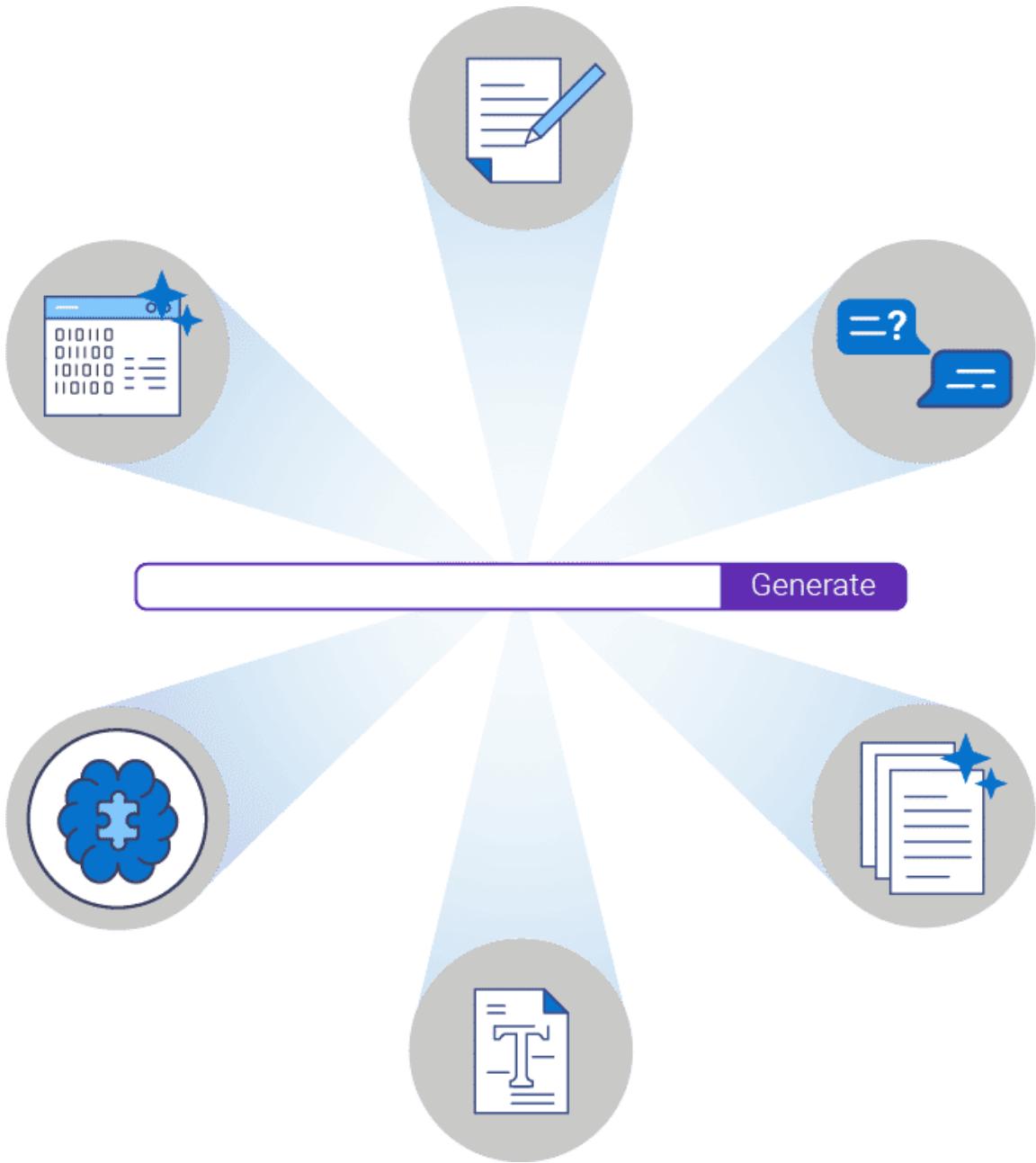
Refer to course for sample Incident report.docx file



Experiment with the prompts in this section. Pay attention to how the output differs depending on the type of prompt.

Challenge activity: Create your own Zero-shot, Few-shot, and Chain-of-Thought prompts.

Application-Specific Prompts



In this section, we will explore examples of prompts that are tailored to common real-world use cases, such as content writing and text classification. While specific prompts are provided for each task, keep in mind that there are multiple ways to accomplish the same objective; the methods shown here are just one of many effective approaches.



Note: In the examples below we have also included prompts to showcase Retrieval Augmented Generation (RAG) discussed in the first module of the course.

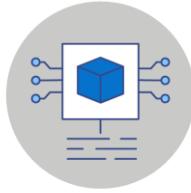
Content Writing

Generative AI can be applied to various content writing tasks, including:

- Rewriting or enhancement
- Translation
- Grammar and style correction
- Summarization

Below are example prompts that are aligned with different prompt engineering techniques we have discussed, customized for specific content-related use cases.

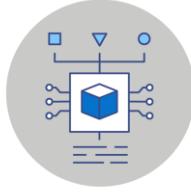
Examples:



Zero-shot

Prompt: “Translate the following text into Hindi: [text].”

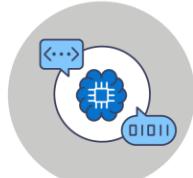
Note: *No examples or additional context are needed—ideal for straightforward language tasks.*



Few-shot

Prompt: “Summarize the following text using the provided examples: [example_1], [example_2], [example_3].”

Note: *Helps guide tone, structure, and level of detail by showing the model what a good summary looks like.*



RAG
Retrieval Augmented Generation

Prompt: “Generate a summary that combines relevant information from the provided [text summaries] retrieved from [external sources]. Check for grammar and consistency in the generated summary.”

Note: *Ideal when the model needs to synthesize information from outside documents.*

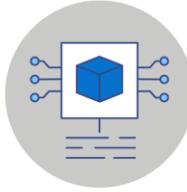
Question Answering

Generative AI can help in various question-answering tasks, especially for:

- Follow-up questions
- Direct queries, MCQ generation
- Information extraction questions

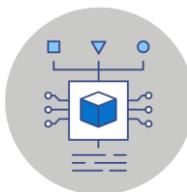
Here are some example prompts for the engineering techniques we have discussed, customized for specific applications.

Examples:



Zero-shot

Prompt: "Answer the following question:
[question]."



Few-shot

Prompt: "Answer the following question:
[question] based on the provided examples:
[example_1], [example_2], [example_3]."



RAG
Retrieval Augmented
Generation

Prompt: “What are the possible questions arising from the provided [text summaries] retrieved from [sources].”

Text Classification

Generative AI can help in various text classification tasks, especially for:

- Analytical work
- Decision making

Below are example prompts that are aligned with specific prompt engineering techniques, which are tailored for classification scenarios.

Example:



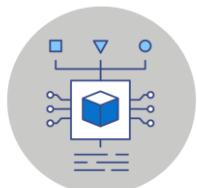
RAG
Retrieval Augmented
Generation

Prompt: “Based on the following reference texts from [trusted sources], classify the input into the correct category.

Reference set: [retrieved_text_1],
[retrieved_text_2]

Input Text: [input_text]”

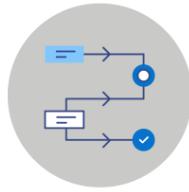
Note: Useful when classification depends on external knowledge or context, like compliance guidelines or domain-specific references.



Few-shot

Prompt: “Classify the following text into the appropriate category using the provided examples: [example_1], [example_2], [example_3].”

Note: Helpful for ensuring that the model follows a consistent pattern based on known examples.



Chain-of-
Thought

Prompt: “Classify the given text into the appropriate category by thinking step-by-step.
Step 1: Identify key terms.
Step 2: Match them with category definitions.
Step 3: Assign the most appropriate label.
Text: [input_text]”.



Tip: For best results, ensure that each classification example is clearly labeled, distinct from others, and aligned with the same format as the input text.

Generating Content

Generative AI can support various creative and professional writing tasks, including:

- Writing engaging social media posts, email campaigns and product descriptions.

- Summarizing research papers, generating reports, and creating data visualizations.
- Managing schedules, drafting email messages, and organizing tasks.

Below are examples of how different prompt engineering techniques can be applied to content generation scenarios.

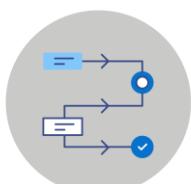
Examples:



RAG
Retrieval Augmented Generation

Prompt: “Based on the provided knowledge, generate an [email] for [audience] that incorporates relevant details from the knowledge source. Specify which information should be retrieved and how it should be integrated into the email.”

Note: Use this when generating content that requires accurate information from specific documents, policies, websites, or customer data.



Chain-of- Thought

Prompt: “Generate a promotional social media post by considering the following logical steps: [step_1], [step_2], [step_3].”

Note: Use this when reasoning is required before generating content (e.g., analyzing sentiment, tone, or domain-specific language).



Tip: For best results, ensure that your prompt includes clear instructions, a defined audience, and constraints like tone or length when needed (for example, “Keep it under 100 words” or “Use a formal tone for executives”).

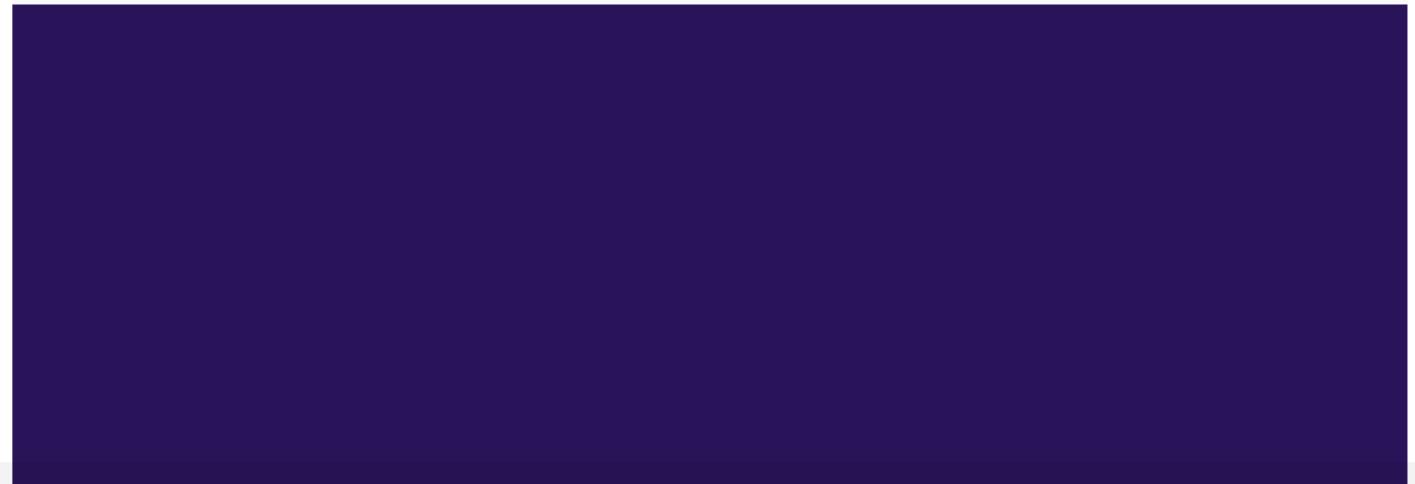
Logical Reasoning

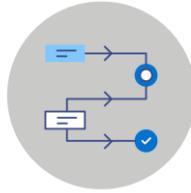
Generative AI can assist with tasks that require step-by-step reasoning, such as:

- Mathematical proofs
- Causal chains of events
- Hypothesis evaluation
- Ethical reasoning or trade-off analysis

Each prompting technique can support reasoning differently, some by scaffolding logical steps, others by referencing known examples or pulling relevant knowledge.

Examples:





Chain-of-Thought

Prompt: "Think step-by-step and provide a logical reasoning process for the given problem:

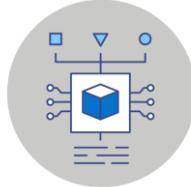
'If a factory increases its output by 10% each month, how much will it produce after 3 months starting from 100 units?'

Include the following steps:

Step 1: Identify the initial value.

Step 2: Apply the percentage increase each month.

Step 3: Calculate the final output."



Few-shot

Prompt: "Provide a logical reasoning step for the given problem based on the provided examples: [example_1], [example_2], [example_3]."

Note: Useful in steering models to an expected type of response.



RAG
Retrieval Augmented
Generation

Prompt: “Using the provided background documents, explain the chain of events that led to the 2008 financial crisis.

Incorporate key events from [doc_1], [doc_2], and [doc_3] to support your answer.”

Note: *Useful for historical or policy reasoning, where external knowledge is required.*

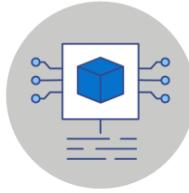
Code Generation

Generative AI can assist with:

- Code creation (writing functions, scripts, or components)
- Code inspection (explaining or reviewing logic)
- Code fixing (debugging, refactoring, or optimizing)

Different prompt engineering techniques support different levels of complexity and context.

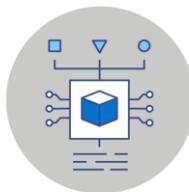
Examples:



Zero-shot

Prompt: “Generate a MySQL query to get a list of all students who have passed. Table name: students, Columns: id, name, score, status”

Note: *Best for quick, simple code snippets without needing prior examples.*



Few-shot

Prompt: “Here are some example inputs and outputs for Python list operations:

Example 1: Input: [1, 2, 3] → Output: [3, 2, 1]

Example 2: Input: ['a', 'b', 'c'] → Output: ['c', 'b', 'a']

Now reverse the following list: [10, 20, 30, 40]”

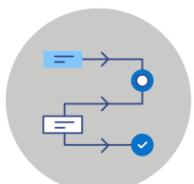
Note: *Helps when the model needs to learn formatting, logic patterns, or syntax style.*



RAG
Retrieval Augmented Generation

Prompt: “Based on the provided documentation for the Pandas library, write a Python function that filters a DataFrame to include only rows where ‘status’ is ‘Active’ and ‘score’ > 70.”

Note: *Ideal when code must conform to external libraries, APIs, or custom specifications.*



Chain-of- Thought

Prompt: “Step 1: Create a list of 10 student names.

Step 2: Create a list of percentage scores for those students.

Step 3: Combine the two lists into a JSON object, where each student is paired with their score.”

Note: *Use when generating multi-step logic or code involving data transformation or structuring.*

Professional Practices for Prompt Engineers

Now that we have grasped engineering prompts, it is time to explore professional practices to further improve prompts.

Improving the Performance of Prompts

Based on the output you need, considerations for prompts vary. It is important to keep these considerations in mind when you are creating prompts for specific applications.

Considerations for text, code, and image prompts.

KEY CONSIDERATIONS FOR EFFECTIVE TEXT PROMPTS	KEY CONSIDERATIONS FOR EFFECTIVE CODE PROMPTS	KEY CONSIDERATIONS FOR EFFECTIVE IMAGE PROMPTS
Incorporating relevant context and metadata within text prompts can render them highly effective. Metadata provides information about other data, such as the origin of the content, when the content was created, and the format of the data. It enhances data usability by making it easier to search, understand, and manage. Example questions to infer metadata: <ul style="list-style-type: none">• What metadata is essential for identifying the author of a book?		

- What metadata can be used to track the date and location where a photo was taken?
- What metadata is necessary to identify the artist and album of an audio track.

Example Prompt: "Provide detailed information about Beethoven's Symphony No. 9 in D minor, Op. 125."

KEY CONSIDERATIONS FOR EFFECTIVE
TEXT PROMPTS

KEY CONSIDERATIONS FOR EFFECTIVE
CODE PROMPTS

KEY CONSIDERATIONS FOR EFFECTIVE
IMAGE PROMPTS

- **Detailed Error Description:** Clearly describe the error or bug, including any error messages, symptoms, and the part of the code where the issue occurs.
- **Relevant Code Snippets:** Provide the relevant sections of code that are causing the problem. This helps the AI focus on the specific area that needs debugging.
- **Environment Information:** Include details about the development environment, such as the programming language, framework, and version being used.
- **Expected Behavior:** Explain what the code is supposed to do versus what it is actually doing. This helps in identifying discrepancies and finding solutions.
- **Previous Attempts:** Mention any debugging steps already taken and their outcomes. This prevents redundant suggestions and helps the AI build on previous efforts.

KEY CONSIDERATIONS FOR EFFECTIVE
TEXT PROMPTS

KEY CONSIDERATIONS FOR EFFECTIVE
CODE PROMPTS

KEY CONSIDERATIONS FOR EFFECTIVE
IMAGE PROMPTS

- **Descriptive Detail:** Provide detailed descriptions of the desired image elements, such as objects, colors, backgrounds, and composition.
- **Style Specification:** Clearly specify the visual style you want, using terms like "photorealistic," "cartoon," "impressionist," etc.
- **Context and Purpose:** Explain the context and purpose of the image. For example, is it for marketing, educational purposes, or artistic expression?

- **Reference Images:** Include reference images if possible. These serve as visual guides and help the AI match the desired style and elements more closely.
- **Technical Specifications:** Provide technical details such as resolution, aspect ratio, and file format.
- **Keywords and Phrases:** Use specific keywords and phrases that capture the essence of what you want.

It is crucial to stick to effective practices and steer clear of common pitfalls. This ensures better results and helps avoid unnecessary complications. Below are some of the general best practices in prompt engineering.



Models

To attain optimal results, it is advisable to use the latest and most advanced models.

Delimiters

Use delimiters to clearly indicate distinct parts of the input. Use delimiters such as, “““, < >, <tag> </tag> to distinguish different sections of the input.

Instructions

Provide instructions on what to do, rather than what not to do. Be specific and descriptive about the required context, outcome, length, format, style, etc.

- Use active voice to make prompts clear and direct.
- Avoid vague descriptions.
- Incorporate specific keywords to narrow down the focus.
- Define a persona to tailor the prompt to a specific audience or context.

Examples

Provide examples to get the desired result.

- The model responds better to specific format requirements shown through examples.
- This approach also simplifies the process of parsing multiple outputs programmatically.

Allow the model time to think

Breaking down complex tasks into smaller steps can be beneficial.

Instruct the model to work out its solution before jumping to a conclusion.

Specific Code

Use leading words for code generation

- This might include using words like ‘import’ to hint to the model that it should start writing in Python.
- ‘SELECT’ for initiating a SQL statement.



Prompt Design

Scenario: You are designing a prompt for a Customer Support chatbot for a healthcare platform. The chatbot must assist users in scheduling appointments, providing medication reminders, and answering basic health-related FAQs.

Using the same scenario, reflect on what you have learned so far. Open the first tab below for the reflection question. When you are ready for the answer, open the second tab.

Reflection Question

You've designed the following prompt based on your prompt plan: "You are a healthcare assistant AI for HealthX. When a patient asks to schedule an appointment, respond with available dates and times for their primary care physician. If no slots are available, direct the user to customer support."

Which prompt design best practices did you apply?

Answer —

Design Best Practices Applied:

- Clear role assigned
- Explicit task described
- Constraints included
- Intended audience clearly identified

Summary

We have put into practice what we have learnt so far and also looked at refining prompts using techniques available. Remember! Practice makes perfect. So, keep practicing until you command the art of Prompt Engineering.

Knowledge Check

Let us recap our understanding of Prompt Design with a quick knowledge check.

What is the mistake in the prompt 'Explain the theory of relativity in one sentence'?

- Lack of clarity
- Lack of context
- Overly ambitious

SUBMIT

How can you avoid 'Lack of Context' in prompts?

- Be specific about what you want LLM to do.

Provide enough background information for LLM to understand the



task.



Set realistic expectations.

SUBMIT

Legal and Ethical Compliance in Prompts

As with any powerful technology, AI has the potential to drive innovation or cause harm depending on how it is applied. Ensuring responsible prompt design means considering fairness, accountability, and safeguards from the start, so that AI systems operate safely, ethically, and in line with regulations. Let us explore some of the legal and ethical aspects in prompt engineering.

Legal Considerations for Prompts

Poorly designed prompts can lead to irrelevant, inaccurate or even harmful AI output, posing significant risks to confidentiality, integrity, and compliance. In order to use AI effectively, ensure that the prompts and resulting outputs are legally sound.

In this regard, here are some key areas to focus on:

Copyright

Copyright law grants creators of original works exclusive rights to reproduce, distribute, publicly perform/display, and create derivative works.

Examples of Potential Infringement by Generative AI:

- **Text:** Prompts that generate articles, stories, or poems may result in content that closely resembles existing copyrighted work.
- **Images:** AI-generated visuals may infringe on the rights of original artworks if they are substantially similar.
- **Code:** Prompts that generate code can lead to infringement if the output replicates proprietary software.
- **Music:** AI-generated tracks that mimic existing songs could violate copyright protections.

Trademark

A **trademark** can include words, phrases, logos, sounds, or even gestures that distinguish one brand's goods or services from another. Trademark law protects the goodwill and reputation of brands.

Examples of Potential Infringement by Generative AI:

- Using prompts to create marketing campaigns that include trademarked slogans or logos can lead to trademark infringement.

Privacy law

Privacy laws protect individuals' personal information from unauthorized collection, use, or disclosure. These laws vary by country but often regulate how data is gathered, stored, and shared.

Examples of Potential Infringement by Generative AI:

- Prompts used in customer service applications may inadvertently expose personal details (e.g., financial or medical data), leading to privacy violations.

Contract law

Contract law governs the formation and enforcement of agreements between parties. It ensures that obligations made in exchange for something of value are legally binding.

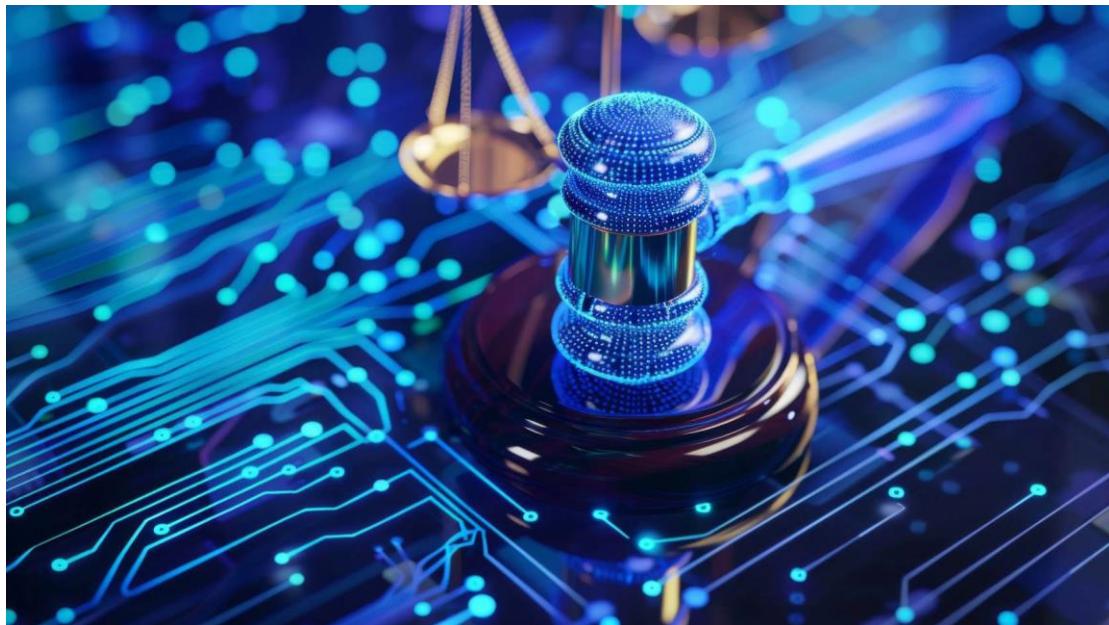
Examples of Potential Infringement by Generative AI:

- Prompts used in AI systems might unintentionally accept terms and conditions on behalf of users without their explicit consent, raising questions about contract validity.

Remember, you are engaging with an AI model to generate data, this data can in-turn be used in business and professional space. It is important to understand the legal implications of using such data. For example: if you create training using generative AI, what are its implications on the IP of the training?

Navigate the below content for detailed understanding of legal obligations.

How Does Copyright Law Apply to Generative AI Content



- Copyright law traditionally requires human authorship for a work to be eligible for protection. As a result, **content created entirely by generative AI—without human input—may not qualify for copyright.**
- However, if a human contributes meaningfully to the creative process, such as by crafting detailed prompts, curating outputs, or editing the results, the final work may meet the threshold for copyright protection.

What are the Ownership and Attribution Complexities

Determining ownership of content produced by LLMs can be complicated, especially when multiple parties are involved in the creation process. Ownership disputes may arise between:

- **Developers of the LLM** who built or trained the model.
- **Users who provided the prompts** that guided the output.
- **Third parties whose data may have been included** in the model's training set.

These overlapping contributions make it difficult to assign clear and exclusive ownership rights.

Attribution and Avoiding Plagiarism

- **Attribution:** Ensure all contributors are acknowledged appropriately. This includes the AI system, prompt designers, and any human editors or collaborators involved in refining the output.
- **Avoiding Plagiarism:** Use plagiarism detection tools to verify the originality of AI-generated content. When applicable, disclose AI involvement and avoid presenting generated output as wholly original human work.

What are the Trademark Risks in AI-Driven Branding and Marketing

Trademarks protect brand identity and help distinguish products and services in the market.

When using generative AI to create branded content:

- It is essential to carefully consider potential legal implications and avoid infringing on existing trademarks.

Risks and best practices when using generative AI for branding:

- **Risks:** There is a risk of trademark infringement if AI-generated content includes trademarked names, logos, taglines, or other protected brand elements without proper authorization.
- **Best practices:**
 - Conduct thorough trademark searches to ensure that generated content does not unintentionally replicate protected brand assets.
 - Obtain necessary permissions before using AI-generated content in any public-facing branding or marketing campaign.
 - Implement prompt engineering guidelines to steer AI away from referencing known trademarks or brand names.

How Does Generative AI Produce Defamatory or Libelous Content

Large Language Models (LLMs) can generate false statements that may harm an individual's or organization's reputation. This is referred to as defamatory or libelous content.

- There have been documented instances where AI-generated content has led to defamation lawsuits. These cases underscore the importance of oversight when using AI to generate public-facing content.

Legal implications of publishing or distributing defamatory content:

- Legal consequences can include **lawsuits, financial penalties, and reputational harm**.
- **Defamation laws vary by jurisdiction**, which affects both how defamation is defined and the severity of potential penalties.

Legal defenses and mitigation strategies:

- Common defenses against defamation claims include:
 - **Truth:** If the statement is factually accurate, it is not considered defamatory.
 - **Opinion:** Expressing subjective opinion (clearly labeled as such) is generally protected.
 - **Consent:** If the allegedly defamed party consented to the publication, liability may be reduced or eliminated.
- **Mitigation Strategies:**
 - Include disclaimers when using AI-generated content, clarifying that outputs may not be verified.
 - Implement content verification and human review processes before publication to minimize risk.

How Do Terms of Service Impact the Legal Usage of Generative AI

Terms of service (ToS) outline the contractual obligations under contract law for using large language model (LLM) APIs. These agreements often include usage restrictions, liability clauses, and other legal conditions that govern how the services can be accessed and deployed.

- Understanding these terms is essential for ensuring legal compliance and minimizing business and reputational risks.

Key Clauses to Review in API Agreements:

- **Usage Restrictions:** ToS documents frequently define permitted use cases and explicitly prohibit certain types of applications (e.g., illegal content generation, misinformation, or use in restricted regions). Violating these restrictions can result in account suspension or legal action.
- **Liability:** These clauses outline the responsibilities and limitations of both the service provider and the user, especially in the event of system errors, misuse, or damage resulting from AI-generated content.
- **Jurisdictional Differences:** Terms may specify which country's laws apply to the contract and how disputes will be resolved. Legal obligations can vary significantly based on regional regulations.



Go To: View [The legal issues that are presented by generative AI](#) article for more information about cases involving Generative AI lawsuits.

This video provides context on the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), and how to protect Personally Identifiable Information (PII).



Best Practices for Ensuring Legal Compliance

Consider, for example, if you request Copilot to create an image of a superman, it might not generate due to copyright restrictions. However, other AI tools might generate the image. This highlights the importance of understanding different

contexts and ensuring compliance to avoid legal issues and maintain trust with clients and stakeholders.

Understanding legal compliance is critical to protect your organization's data.

Some compliance best practices



Adhere to the terms of service for LLM platforms and APIs. Understand usage restrictions, licensing terms, and liability clauses before integrating AI outputs into products or services.



Do not use copyrighted material in prompts, training data, or outputs unless you have obtained appropriate permissions or licenses.

2 of 5



Stay informed about updates to intellectual property, data protection, and AI-specific regulations across relevant jurisdictions.

3 of 5



Implement anonymization and data de-identification practices when handling personal or sensitive information, especially in customer-facing applications.



Regularly monitor AI-generated brand names, logos, and taglines to ensure they do not infringe on existing trademarks or misrepresent your brand.

5 of 5

As AI systems become more influential, ethical prompt design is critical to prevent bias, ensure fairness, and promote accountability. Misleading or biased prompts can reinforce harmful stereotypes, while unclear responsibility for AI-generated content complicates the ethical landscape. Ethics in prompt is about ensuring fairness, respect, and inclusivity in the way we design and use them.

Select the slide deck to explore more on how to achieve ethical prompts under key areas of concern.

Bias and Discrimination



Bias in the context of AI prompts refers to the tendency of AI systems to produce outputs that reflect prejudices or stereotypes present in the training data.



Discrimination in AI prompts occurs when these biases result in unfair treatment of certain groups.

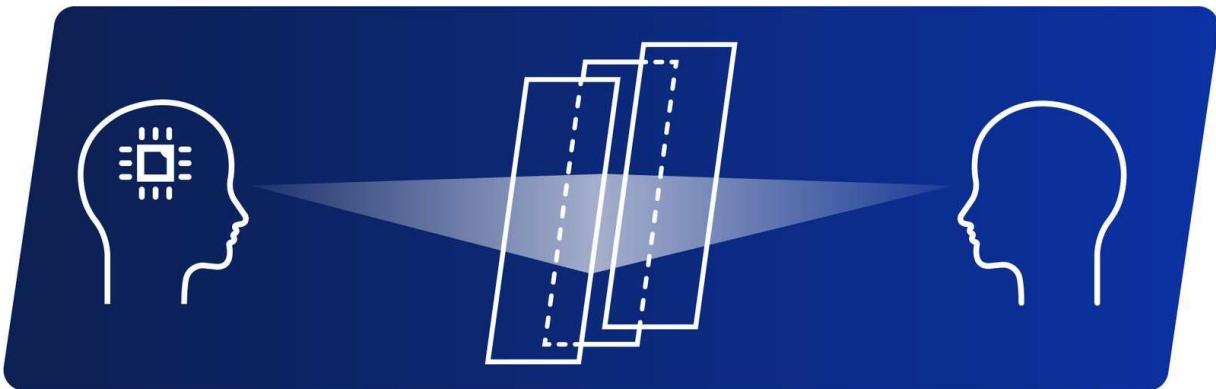
Prompts can **unintentionally introduce bias** into AI-generated content. If a prompt contains biased language or assumptions, the resulting output **may reinforce harmful stereotypes or marginalize certain groups**.

- For example, a prompt like "Why are women weaker at sports?" can generate responses that perpetuate gender-based stereotypes.

To mitigate this risk:

- Design prompts that are neutral and inclusive.
- Use diverse datasets that reflect a broad range of perspectives and lived experiences.
- Continuously evaluate outputs for signs of implicit bias and adjust prompts or models accordingly.

Transparency and Accountability



Transparency in prompt-based AI systems is essential for building trust and ensuring ethical use.

- Users should be clearly informed when content is generated by AI in response to prompts, and there should be defined processes for reviewing and approving this content.
- Accountability mechanisms, such as audit trails, version histories, and user feedback systems, help ensure that AI-generated outputs meet ethical standards and can be traced back to their source for verification or correction.

Human Dignity and Autonomy



Prompts can be used to persuade or manipulate, which raises **significant ethical concerns**, particularly when they influence users' thoughts, emotions, or behaviors.

Because prompts have the power to generate highly convincing content, it is crucial to design them in ways that respect human dignity and uphold individual autonomy.

To support ethical use:

- Avoid manipulative or coercive language in prompt design.
- Integrate human oversight to review and approve AI-generated content.
- Ensure that all outputs align with ethical standards and respect users' rights, freedoms, and decision-making ability.



A model performs as it was programmed to do. Usually training process has a step that is called 'fine-tuning' intended to minimize bias. However, some use-cases have a higher risk of facing these issues than others.

Guidelines for Designing Unbiased Prompts and Comply with Ethics

- 1 **Use Gender-Neutral Language:** Opt for inclusive terms like "person" instead of "man" to avoid reinforcing gender stereotypes.
- 2 **Diversify Example Perspectives:** Include a variety of cultural, social, and economic backgrounds in prompts to expose the AI to diverse viewpoints.
- 3 **Avoid Stereotypical Roles and Attributes:** Present characters and scenarios that do not conform to traditional stereotypes to reduce biased outputs.
- 4 **Implement Contextual Details:** Provide relevant context in prompts to help the AI understand nuances and generate accurate, fair responses.
- 5 **Conduct Regular Audits:** Systematically review AI outputs for fairness and inclusivity to identify and rectify biases.
- 6 **Engage Diverse Teams:** Involve individuals from various backgrounds in the prompt design process to recognize potential biases.

7

Feedback Mechanisms: Establish channels for user and stakeholder feedback to identify biases and improve prompt design over time.

Developing Ethical and Legal Guidelines

In this video, we discuss how to translate legal and ethical constraints into practical guidelines, to ensure everything stays compliant. Let us get started!





Evaluate ethical risks or compliance gaps in your prompt.

Scenario:

You are designing a prompt for a Customer Support chatbot for a healthcare platform. The chatbot must assist users in scheduling appointments, providing medication reminders, and answering basic health-related FAQs.

Using the same scenario, reflect on what you have learned so far. Open the first tab below for the reflection question.

When you are ready for the answer, open the second tab.

Reflection Question —

In your prompt:

- Is sensitive information (e.g., medical records) protected?
- Will the responses be respectful and inclusive?
- Could the AI give misleading or unauthorized responses?

Answer

Sensitive Information: Ensure the AI doesn't show another patient's data or give medical advice beyond its scope.

Updated Prompt to protect sensitive information: "You are a healthcare assistant AI for HealthX. When a patient asks to schedule an appointment, respond with available dates and times for their primary care physician. If no slots are available, direct the user to customer support. Only display information related to the logged-in patient's own appointments. Do not provide details on any other patient's data. If the system cannot access appointment data, respond politely and refer the user to human support."

Respectful Language: The prompt is role-based. With the AI acting as a health care assistant, it should use respectful and inclusive language.

Accuracy: The AI is instructed to direct the user to customer support if no appointments are available. This should prevent fabricated answers.

Summary

We must understand that while the future of prompt engineering comes with its fair share of challenges, it also brings a wealth of exciting opportunities. By adapting to legal frameworks and adhering to ethical considerations, we can use prompts in a way that benefits everyone. This approach ensures that we harness the power of AI responsibly and effectively, paving the way for innovative and legally sound applications.

Knowledge Check

Let us recap our understanding of Legal and Ethical Considerations with a quick knowledge check.

Imagine you are part of a team designing prompts for an GenAI model. Your team is tasked with ensuring the prompts are unbiased and ethical. During a review, you notice that a prompt consistently portrays a specific gender in a stereotypical role. What action should your team take to address this issue?

- Add more stereotypical roles to balance the prompt
- Use gender-neutral language to avoid reinforcing stereotypes
- Remove all contextual details from the prompt
- Limit feedback mechanisms to internal team members only

SUBMIT

Which of the following is a potential copyright infringement by generative AI?



Generating a new programming language



Creating AI-generated visuals that are substantially similar to existing artworks



Using prompts to create marketing campaigns with original slogans



Collecting anonymous user data for analysis

SUBMIT

Prompt Template Documentation

Prompt engineering does not exist in isolation. It is deeply influenced by the surrounding ecosystem, including the users and their roles, specific use cases, technological context, and real-world applications. Whether you are designing prompts for a chatbot, creating templates for enterprise-wide use, or fine-tuning prompts for specific customer interactions, achieving successful outcomes relies on a shared understanding, collaboration, and structured documentation.

This module elaborates on three key aspects of effective collaboration and communication: prompt template documentation, prompt template documentation format and tools, and communication and collaboration in Prompt Engineering.

Prompt Template Documentation

Prompt template documentation refers to the structured record of how a particular prompt is designed, what it does, and how it should be used or

modified. This documentation ensures that prompts can be reused, improved, and scaled across teams. It standardizes the format and content of prompts to ensure consistency, efficiency, and accuracy in the responses generated by the LLM.

In a Customer Support environment, a prompt template might be:



Prompt: "Provide a detailed response to the customer's issue regarding [User Query]. Include steps for resolution and any additional resources."

This template helps support agents quickly generate comprehensive responses to common issues. Prompt templates are essential tools in prompt engineering, enabling scalable and high-quality interactions with LLMs.

What problem does it solve?

A prompt template solves several key challenges:

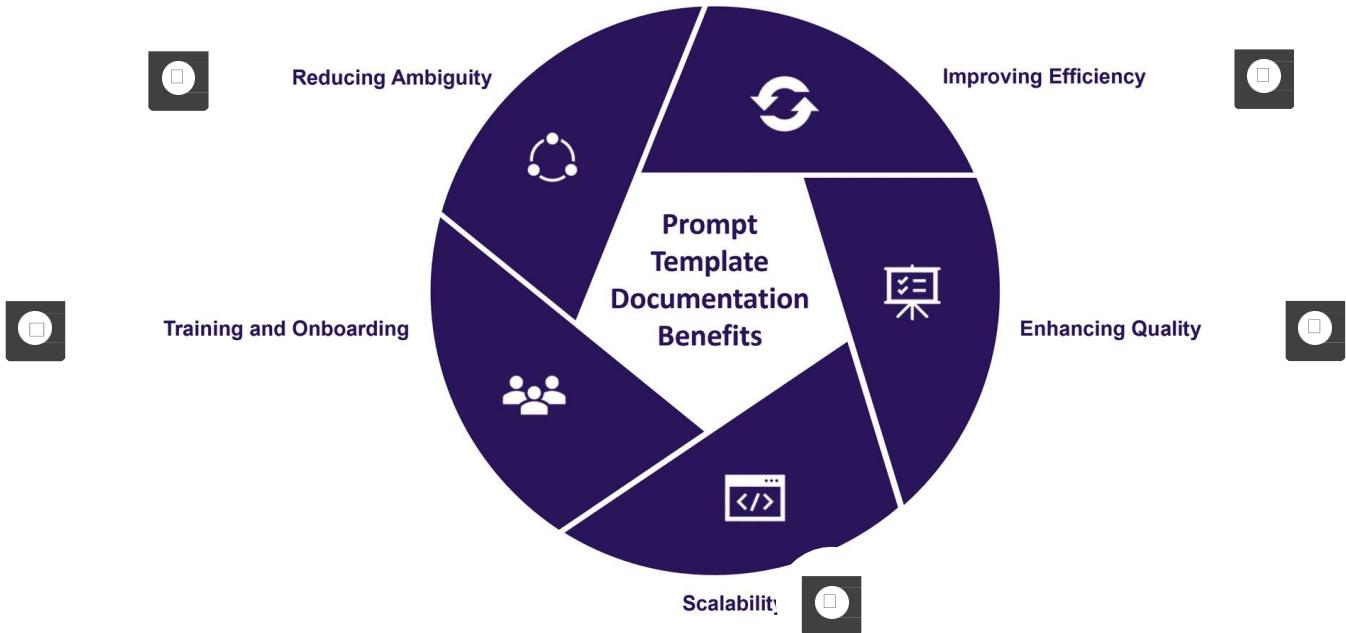
- **Consistency:** Ensures that responses are consistent in structure and style, maintaining quality.

- **Efficiency:** Accelerates content creation, saving time and effort.
- **Guidance:** Provides a clear path for creating responses, reducing ambiguity.
- **Quality Control:** Maintains high content quality through structured format compliance.
- **Scalability:** Supports replication of content across various projects, enabling scalability.

Benefits of Creating Prompt Template Documentation

A prompt template is designed to streamline the process of interacting with language models by providing a consistent structure.

Benefits of using prompt templates.



Reducing Ambiguity

Prompt templates minimize misunderstandings by clearly defining expectations. They offer a structured approach to prompt creation, which can help in achieving more accurate and relevant responses.

Improving Efficiency

Prompt templates save time by providing reusable templates for similar tasks.

Enhancing Quality

Prompt templates ensure high-quality responses by following a proven format. They minimize the chances of errors or misinterpretations by providing a clear framework for prompt creation.

Scalability

Prompt templates facilitate the creation of multiple prompts for different scenarios without starting from scratch each time.

Training and Onboarding

Prompt templates help new users or team members quickly understand how to create effective prompts. They serve as a valuable resource for training and onboarding, ensuring that everyone

follows best practices.



Tip: Well-documented prompt templates become living knowledge assets, improving collaboration, transparency, and innovation across prompt engineering teams.

What Should Good Prompt Documentation Include?

A well-documented prompt template serves as both a technical asset and a communication tool across teams. It ensures that prompts are clear, reusable, and understandable by different stakeholders from developers to compliance teams.

Below is an elaboration of the core fields that every prompt documentation should include, along with real-world context and examples.

Select the + icon to learn more about each core field, their description, and the example.

Template Name

—

A unique and descriptive identifier for the prompt template.

Rewrite_with_tone_friendly, generate_email_summary_v2

Purpose

Explains the business need or problem the prompt is solving.

Input Parameters

List of variables or placeholders used in the prompt, often dynamically filled by user input or system data.

```
{customer_name}, {complaint_text}, {tone}, {product_name}
```

LLM Instructions

Clear instructions or behavior expected from the model, often a system prompt.

```
Rewrite the input text in a friendly and apologetic tone suitable for email.
```

Expected Output

Describes the format, tone, length, or structure of the desired model output.

```
One-paragraph email in professional tone, max 100 words, begins with greeting.
```

Examples or Scenarios of Usage

Illustrative sample inputs and outputs to demonstrate how the prompt works.

Limitations or Known Issues

—

Any known model risks, biases, edge cases, or hallucination tendencies.

LLM might over-apologize or invent resolution steps if instructions are vague.

Version History

—

Tracks changes over time, including what was changed, who made the change, and when.

V1.0 - Created by Ayesha on March 12.

v1.1 - Added tone parameter by Sam on Apr 3.

Related Stakeholders

Teams or individuals involved in design, review, or usage of the prompt.

UX Writers, Legal Team, Product Manager, Customer Service Ops.



Tip: If you are using an API, templates are essential to ensure proper outcomes.



Use an AI model to experiment with the following template. Change the template inputs to see how they affect your output. Use the [Additional Notes](#) document for guidance.

Template: Provide a detailed explanation of [User Query] suitable for [Context]. Respond in a [Tone] manner and structure the information in [Format].

Template Inputs:

- User Query: 'the process of quality control in manufacturing'
- Context: 'a training session for new employees'
- Tone: 'formal'
- Format: 'bullet points'

Activity challenge: Create your own template.

Prompt Template Documentation Format and Tools

Prompt Template Documentation Format and Tools

Effective prompt engineering is not just about designing prompts—it is about maintaining, sharing, and evolving them collaboratively. Proper documentation formats and tooling play a crucial role in enabling clarity, versioning, testing, and cross-team communication.

Recommended Formats

To enable access, collaboration, and version control, consider documenting prompts using one of the following formats:

Format	Best For	Notes
Markdown Files	Git-based workflows, technical users	<ul style="list-style-type: none">• Version-control with Git• Lightweight and readable• Best for engineering teams using GitHub or GitLab

		<ul style="list-style-type: none"> • Easy to filter, sort, tag • Accessible to non-user • Good for content reviews, UX, or marketing teams
Spreadsheets (Excel, Google Sheets)	Cataloging large sets of prompts, stakeholder visibility	<ul style="list-style-type: none"> • Rich-text formatting • Tagging, comments, permissions • Useful for documentation, governance, and cross-team visibility
Notion or Confluence	Cross-functional collaboration	<ul style="list-style-type: none"> • Combine prompt + code + output in one document • Ideal for testing prompts with real-time output
Jupyter Notebooks	Data scientists and prompt engineers	<ul style="list-style-type: none"> • Machine-readable • Ideal for use with tools like LangChain or OpenAI API • Supports automation pipelines and infrastructure-as-code models.
JSON or YAML	Integration with systems, APIs, automation	<ul style="list-style-type: none"> • Machine-readable • Ideal for use with tools like LangChain or OpenAI API • Supports automation pipelines and infrastructure-as-code models.

Recommended Tools

The right tools complement your format and make it easier to collaborate, iterate, and deploy prompt templates in production environments.

Tool	Use Case or Value
Git, GitHub, or GitLab	Version control for prompts, pull request workflow, collaboration using branches, and reviews.
Notion or Confluence	Team-wide documentation platform with tagging, commenting, access control, and approval workflows.
Prompt Layer, LangChain Hub, or OpenPrompt Studio	Specialized platforms for managing, evaluating, and storing prompt templates across versions.
VS Code and Markdown Preview	Lightweight developer IDE setup for editing prompt templates in Markdown format with live preview.
Trello, Jira, or Asana	Organizing prompt engineering tasks in Agile workflows, assigning reviewers, and tracking changes.

CONTINUE

Communication and Collaboration in Prompt Engineering

Communication and Collaboration in Prompt Engineering

Effective communication and collaboration are essential for scaling prompt engineering in real-world environments. As prompt templates evolve in response to user needs, model updates, and business requirements, it is critical to establish structured processes. These processes ensure alignment, consistency, and innovation across teams.

Whether you are part of an AI product team, marketing department, legal review board, or engineering group, prompt engineering success depends on:

- Clearly defined roles and responsibilities.
- Strong communication channels to keep teams aligned.
- Collaborative tools for co-editing, reviewing, and testing prompts.
- Continuous feedback loops for prompt evaluation and iteration.

When these elements are in place, organizations can create prompt systems that are scalable, reliable, and adaptable, all while maintaining compliance, quality, and user experience standards.

Let us now look at an example of collaboration between a legal and prompt engineering team. Navigate the cards below.

Example

Legal Team and Prompt Engineering Collaboration

Step 2

Prompt Team Drafts

"Summarize the customer's transaction history for the last 3 months in plain English."

Step 3

Legal Reviews and Flags Issues

- Over-disclosure of sensitive financial data.
- Language that may imply financial advice.

Step 4

Prompt Team Revises for Compliance

"List the transaction categories from the last 3 months, without disclosing individual amounts or offering financial advice."

- Aligned with legal guidelines.
- Clear in scope and safer in tone.
- Preserves user value without regulatory risk.

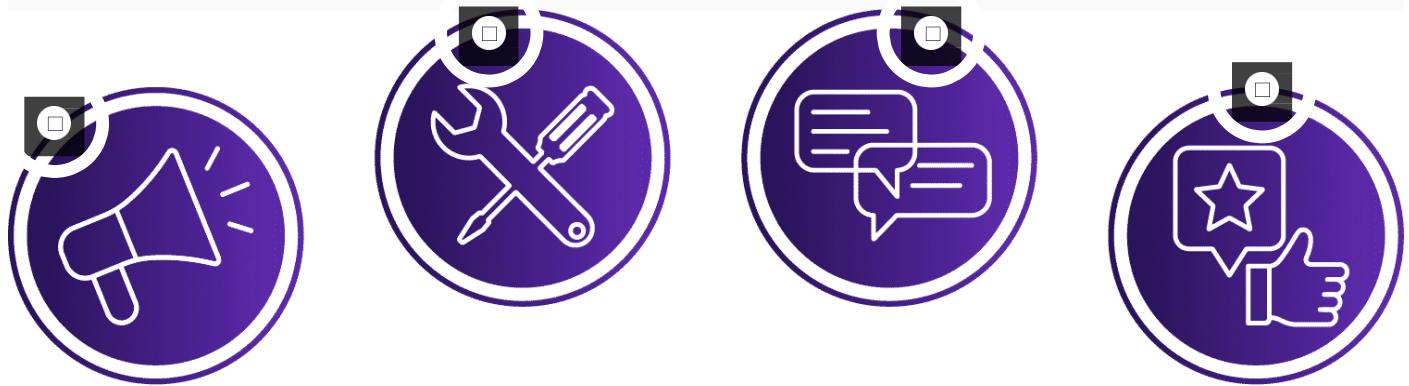
Summary

Legal Team and Prompt Team collaboration and communication are an ongoing process.

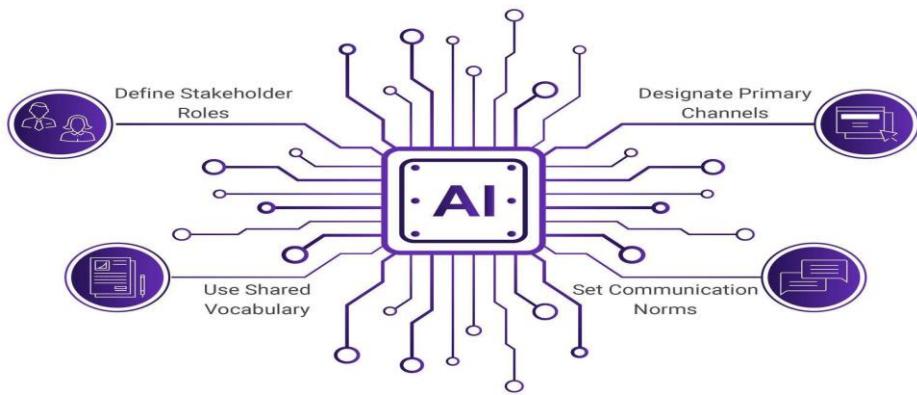
Strategies for Effective Communication and Collaboration

Effective strategies for collaboration and communication in prompt development involve fostering open dialogue among team members, leveraging collaboration tools, conducting meetings and reviews. Providing iterative feedback ensures alignment and creativity.

Strategies for communication and collaboration.



Establishing Clear Communication Channels and Protocols



- To avoid ambiguity and ensure alignment across all stakeholders, including prompt engineers, designers, product managers, legal teams, and developers, it is essential to establish clear communication protocols.
- These protocols help standardize expectations, reduce rework, and ensure accountability at each stage of the prompt lifecycle.

Use Collaborative Tools and Platforms

- Prompt engineering is inherently cross-functional, bridging technical, design, compliance, and business domains.
- A successful workflow requires a well-integrated set of tools that support both asynchronous (examples: documentation, comments, code review) and real-time (examples: meetings, demos) collaboration.
- Examples of tools includes Git, GitHub, Confluence, and Jira. (Please refer to the Recommended Tools topic from previous lesson.)

Conducting Regular Meetings and Reviews

Routine check-ins help teams stay aligned:

- Prompt performance and evaluation metrics.
- Changes in model behavior or API capabilities.
- New requirements or legal concerns.
- Emerging UX or tone issues.

Example: During a weekly review, the prompt engineering team demos updates to a customer support summarization prompt. The legal team flags a potential tone issue, which leads to a revision. The updated prompt is versioned in GitHub and linked in Confluence for cross-team visibility.

Provide and Receive Constructive Feedback

To ensure prompt templates continue to evolve effectively, feedback must be:

- Specific and actionable.
- Collaborative, not critical.
- Backed by rationale or data when possible.

Example: A UX writer suggests clarifying a vague sentence in a prompt. Instead of simply saying "Fix it," they propose a revised version and explain their reasoning. The engineer tests both versions and shares evaluation metrics. The team uses the data to decide on the optimal variant, promoting learning and reducing conflict.



Tip: Establishing a culture of transparent communication, collaborative tooling, and feedback-driven iteration is essential for scaling and maintaining prompt quality across large teams and evolving use cases.

Effective Collaboration Between Various Roles and Responsibilities

This video talks about the effective collaboration between prompt engineers, data scientists, and engineers. It explores the unique perspectives and expertise that each role brings to the table. Finally, it discusses how to leverage the strengths of each team member to achieve outstanding results.





Document and share the final prompt as a reusable template

Scenario: You are designing a prompt for a Customer Support chatbot for a healthcare platform. The chatbot must assist users in scheduling appointments, providing medication reminders, and answering basic health-related FAQs.

Using the same scenario, reflect on what you have learned so far. Open the first tab below for the reflection question. When you are ready for the answer, open the second tab.

Question —

How would you document the final prompt? Include the following in the documentation:

- Use a standard prompt template format (e.g., role | task | input | output).
- Include notes for collaborators (e.g., product owners, data privacy team).
- Create a versioning or feedback section for improvement cycles.

Answer

—

Final Template Example:

- Prompt Name: Appointment Scheduling - Patient
- Role: Healthcare Assistant AI
- Task: Schedule and display available appointment slots
- Input: Patient request (e.g., “Schedule an appointment with my doctor”)
- Output: Available dates and times, confirmation details
- Constraints:
 - Only return data for the logged-in patient
 - Do not fabricate details
- If no data is found, respond:
 - “I couldn’t locate available appointment slots right now. Please reach out to our support team [insert link].”
- Version: v1.2
- Last Updated By: Team L&D - 2025-04-29
- Collaborator Notes: NA
- Data Privacy: Validated with legal and DPO teams
- Future Update: Consider multilingual support
- Feedback Thread: [Insert shared doc or tracking tool link]

Summary

We explored the collaborative nature of Prompt Engineering and the importance of aligning prompts with the broader ecosystem including users, use cases, and technology. We examined how structured documentation, standardized formats and tools, and effective cross-functional communication play a critical role in designing scalable, reusable, and impactful prompts.

By fostering collaboration and clarity, prompt engineers can ensure consistency, efficiency, and success across AI-powered solutions.

Knowledge Check

Let us recap our understanding of Prompt Documentation and Collaboration with a quick knowledge check.

Scenario: You are tasked with creating a prompt template for a customer service chatbot that handles technical support inquiries. The chatbot needs to provide accurate and detailed troubleshooting steps.

Question: Which core field should you include in the prompt template documentation to ensure it addresses technical support inquiries effectively with right tone and format?

- Template Name
- Limitations or known issues
- Expected Output
- Examples or Scenarios of Usage

SUBMIT

Scenario: You are documenting a new prompt template for your team using a Git-based workflow. The template is designed to help support agents quickly generate comprehensive responses to common issues.

Question: Which format would be most suitable for documenting the prompt template to enable easy version control and collaboration?

- Spreadsheets (Excel, Google Sheets)
- Markdown Files
- Notion or Confluence
- Jupyter Notebooks

SUBMIT

Scenario: You are part of a prompt engineering team that needs to ensure alignment and consistency across various stakeholders, including designers, product managers, and legal teams.

Question: Which communication protocol would be most effective for maintaining alignment and reducing rework?

- Ad-hoc meetings as needed

- Weekly status updates via email
- Establishing clear communication channels and protocols
- Using instant messaging for all communications

SUBMIT

Conclusion

We hope you had fun in the journey of understanding AI models, Prompts, how to create them and best practices, the key is to put in to practice what you learned. Remember the model Write > Refine > Test > Iterate and don't forget to have fun while learning.

You have completed this content.

Go to the next learning or assessment, if applicable.

