



Prompt Engineering Beginner

Content:

Conclusion and recap



Generative AI



1 Generative AI is the umbrella of algorithms used to create content ranging from text, image, video, audio, or combinations of above.



2 This is the next step in AI, where content generation is widely used in businesses to solve various problems such as text summarization, advertisement content generation, and many more.



3 Generative AI models such as ChatGPT, DALLE, GPT4 , Mid Journey have gained quite popularity in over the last few years.

Large language models

1

LLMs (Large language models) come under the umbrella of Generative AI, where they cater to delivering textual content using input instructions.

2

These are used to solve various problems on structured and unstructured data such as summarization, question answering, conversational bots etc.

3

Few examples of these LLMs include models such as ChatGPT, Gemini , LLama. These LLMs are foundational models, trained on a broad spectrum of unstructured and generalized data.

What are prompts?

- 1 **Prompts** are input instructions given to generative models in order to direct them to perform a given task.
- 2 Generative models such as LLMs can interact with humans through conversational inputs where users can provide text as input directing the models to perform a given task.
- 3 With systematic and structured prompts, more meaningful output can be generated.

Prompt Engineering

- 1** Prompt engineering bridges the communication gap between humans and Generative models.
- 2** It involves crafting clear and specific prompts or instructions to get the desired output from the model.
- 3** It is the process of formulating prompts or queries that guide the model's responses for a particular task or use case. Prompt engineering is essential to make GPT-3.5 produce accurate, relevant, and useful content.

Benefits of Prompt engineering

- 1 Greater control over outputs
- 2 Improved user experience(can get outputs in desired format as well).
- 3 Easier debugging and structuring of queries & better contextual understanding by models

Few examples:

Basic prompt for summarization:

"Summarize the text given below in triple backticks in 50 words.

``{text}``"

Basic prompt for fetching information :

"Generate a list of top keywords from the given email text delimited by triple backticks.

Provide them in JSON format with the following keys:

email_id, email_content,list of top keywords.

``{text}``"

Content:

Power of LLMs



LLMs

1

LLMs are foundational models trained on huge corpus of textual data captured from the internet.

2

This makes them capable of understanding natural language and perform various generation based tasks on textual data.

3

Earlier textual language models used to be trained on specific tasks and various models with various capabilities needed to be designed to cater to different textual use cases.

LLM Examples

- 1 Chat GPT /GPT 3.5
- 2 BARD/GEMINI
- 3 LLAMA

How are LLMs trained?

- 1 Underlying transformer based decoder architecture with billions of parameters which can handle sequential data like text.
- 2 This decoder architecture has multiple layers of neural networks, each with parameters that can be fine-tuned during training, which are enhanced further by a numerous layer known as the attention mechanism.
- 3 These models learn to predict the next word in a sentence based on the context provided by the preceding words.

How are LLMs trained?

- 1 LLMs are trained on massive corpus of textual data which enables it to understand the context and then perform various tasks through zero shot and self supervised learning.
- 2 Furthermore these foundational models are then fine-tuned with smart techniques such as RLHF which enables it to understand the right context through human feedback in the loop.
- 3 Prompt engineering can then further enable the user to get control of the output generated by LLMs

LLM use cases:

Text generation
-Summarization
- Content generation

Language translation

AI assistants
- Code assistants

Language insights
-Sentiment analysis
-NER

Content:

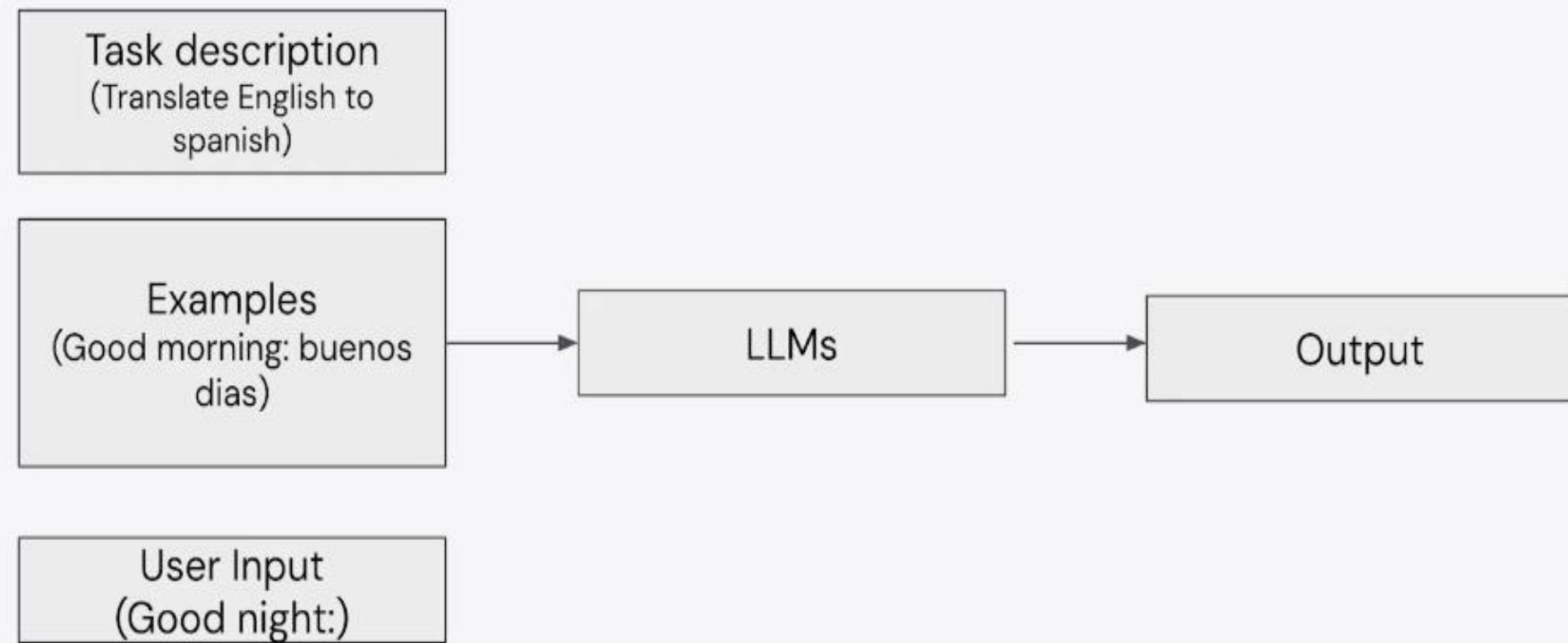
Why use prompts?



Prompts

- 1 Prompts can guide LLMs to generate output in specific direction considering the type, tone, length etc in check.
- 2 These structure the requirements into an input format that effectively communicates the desired outcomes to the model, thereby obtaining the intended output.
- 3 Detailed prompts with clear instructions can cause significant improvements in the quality of outputs generated by LLMs.

Prompt structure



Advantages of prompts

1

They can help generate the output in specific formats adhering to the clear instructions given by the user.

2

Clear instructions and detailed explanation leads to much defined output without leaving much scope for LLMs for their own interpretation.

3

Prompts can also be used for logical follow ups thus ensuring correct output with explained iterations.

Prompt applications

1

LLM results are sensitive to contents provided in the input as they work with next word generation problem. Thus clear prompts with specific instructions help in giving better context.

2

Since LLMs are trained on huge corpus of datasets with their own understanding, one prompt which works for one LLM might not work for another LLM.

3

Small changes in the prompt wordings – even the order of words – can make a significant difference in the quality of the response

Prompt use cases

1

Creative and communications where prompts are used to generate creative content.

2

Personalized education : Acting as AI agent supporting with question answering tasks, or with debugging code.

3

Text insight generation for customer support and review analysis

Prompts- steps

Context

Instruction

Task

Refine

Content:

Building effective
prompts



Prompt structure

- 1 Providing effective context with clear explanation of the background and situation.
- 2 Explain the task in clear and concise words without leaving much room for interpretation for what the model needs to do.
- 3 Giving examples to explain the requirements.
- 4 Give output formats to define the clear format of returning the required text.
- 5 Refining the prompts through multiple iterations.

Providing context

1

When we provide effective context giving clear background, it helps the model understand the situation in which the answers are to be generated.

2

For example if we are to build an AI assistant to automatically take orders in a PIZZA shop, we should be providing the context of the pizza shop, explaining the types of pizzas we deliver, price ranges, timings and more.

3

This is usually delivered as system prompts.

Explaining the task:

1

The language of the request component typically took the form of a question, implicit request, command, or incomplete sentence.

2

For example explaining the task as:

'I want you to act as an executive who can take pizza orders from customers. You should ask them to select pizza they want from the menu only, confirming the price and asking for any additional toppings. Also check if they want to have it delivered and confirm their home address.'

Giving examples

- 1 Providing examples on how to respond in case of certain queries allows the model to understand the scenario and structure of responses it should be delivering.
- 2 Examples can be about the tone and format the answer should be delivered in , or about specific type of response.
- 3 Few shot prompting with some examples is a famous prompting technique.

Defining the output format

1

Output format plays an important role in getting the required answer. We can clearly specify the length of the response, tone(professional/ funny), format of the answer(bullet points or 2 line story) and more.

2

Further if the details generated need to be saved in a csv, the output can also be captured in a tabular format.

3

Sharing the details and examples on output format brings significance improvement in prompts outputs.

Refining the answers:

- 1 Through multiple iterations, prompts can be refined by checking where the model is not performing well.
- 2 Even model can be prompted to ask clarifying questions to get better understanding of the context.
- 3 Examples can be provided to explain the model on what went wrong and how to improve responses on those.

Prompt engineering

- 1 Prompt engineering is the art of designing and refining prompts to get the desired output from LLMs.
- 2 It requires a blend of creativity, understanding of the model's capabilities, and strategic structuring of the question or statement to guide the model towards providing accurate, relevant, and useful responses.
- 3 This improves quality of output, gives better user experience and delivers enhanced results.

Prompt engineering



Be as specific and clear as possible: For input: Giving clear context, scope, constraints for the response. For output: Giving format, tone, length of output.



Less effective: "Tell me about AI."

More effective: "Explain the key principles of artificial intelligence and its applications in healthcare."



Giving clear explanations on how to perform a given task (for example if it is a mathematical calculation, explaining the steps clearly to do so).

Few tips to follow:

- 1 Giving instructions on what to do instead of what not to do such that instead of asking it to not deliver long sentences, asking it to give precise summary in 50 words.
- 2 For example for this basic prompt: You are an agent that recommends movies to a customer. DO NOT ASK FOR INTERESTS OR FOR PERSONAL INFORMATION.
- 3 This could be a better prompt: You are an agent that recommends movies to a customer. You are responsible to recommend a movie from the top global trending movies. You should refrain from asking users for their preferences and avoid asking for personal information.

Few tips to follow:

- 1 Giving the persona or frame to the model to make it understand its role, for example if you are building a bot to assist with code, you can specify if it needs to support with backend, frontend, test or some other role.
- 2 Split complex tasks into simpler ones and asking the model to solve it in a given order and format.
- 3 Specifying the output format with explanation and examples are also suggested.

Key considerations for Prompt engineering

- 1 **Clarity:** Ensuring that your prompt is clear and easy to understand.
- 2 **Contextual information:** When relevant, include background information or context in your prompts. This helps the model understand the task or question better.
- 3 **Specificity:** Provide specific details and context in your prompts, so the model has the necessary information to generate a relevant response.
- 4 **Desired format:** Specify the format or structure you want the answer in, whether it's a paragraph, list, table, or any other desired output format.

Few tips to follow:

- 1 **Tone and style:** If you have specific requirements for the tone, style, or level of formality in the response, make those expectations clear in your prompt.
- 2 **Examples and demonstrations:** You can provide examples or demonstrations in your prompt to illustrate the kind of response you're looking for.
- 3 **Review and iterate:** After receiving responses, review them to ensure they meet your expectations. If needed, iterate and refine your prompts to improve the model's performance.
- 4 **Ethical considerations:** Ensure that your prompts adhere to ethical guidelines and avoid generating harmful, offensive, or inappropriate content.

Content:

Prompt examples



Prompt examples

Task1: Act as a story teller:

I want you to act as an advertiser. You will create a campaign to promote the mentioned product. You will be given the target audience and you need to develop key messages and slogans for the campaign. Additionally will give you the media for the campaign and you need to come up with the activities to reach the target audience.

My first request is "I need help creating an advertising campaign for a fast fashion clothing brand targeted for Genz".

Prompt examples

Task2 : Act as a UI/UX developer:

I want you to act as a UX/UI developer. I will provide some details about the design of an app, website and it will be your job to come up with creative ways to improve its user experience. This could involve creating prototypes, testing different designs and providing feedback on what works best. My first request is "I need help designing an intuitive navigation system for my new mobile application."

Prompt examples

Task3 : Create a how to guide

You are an expert teacher. Create a 10-step guide to go on a 2 day long trek in the mountains. Explain each step in all the detail I will need to be able to do it, in simple terms. Begin with a section listing any items, tools or equipment I will need. If there are any important safety considerations, include a section listing these at the end of the guide. Also include a section on the most common mistakes that beginners will make when attempting this task and how to avoid them.

Use cases in real world: Social Media

1

How can I, as a business owner or social media marketer, leverage social media to increase brand awareness and reach?

2

As a social media marketing manager, how do I respond to people that are writing negative things about my products on Twitter?

3

Write a Facebook bio for my business in California that provides AI services to clients especially in the field of computer vision.

Use cases in real world: Marketing

1

In less than 75 words, generate an email headline that highlights the unique selling proposition of my brand, Starbucks.

2

What campaign should I run in the festive season for my brand which sells ethnic clothes for women to reach the target audience of ages 20-50.

3

Give me an idea for an email promotion in which we offer our customers X% off our product.

Use cases in real world: Content generation

- 1 Can you proof read my blog on 'Himalayan treks' to change the tone to adventure and correct any grammar mistakes.
- 2 Write a compelling intro for a landing page advertising my [type of business] business's [offer + any relevant offer details].
- 3 Write an article on current GDP trends considering the socio economic nature of world.

Few tasks and their sample prompt:

1

Question Answering:

Task: Ask the model to answer a specific question.

Example Prompt: "Can you explain the causes of climate change in a concise manner?"

2

Content Generation:

Task: Request the model to generate content on a particular topic.

Example Prompt: "Please provide an informative article about the benefits of renewable energy sources."

3

Summarization:

Task: Instruct the model to summarize a lengthy article or document.

Example Prompt: "Summarize the key points from the 10-page report on the impact of artificial intelligence on the job market."

Few tasks and their sample prompt:

1

Language Translation:

Task: Translate a sentence from one language to another.

Example Prompt: "Translate the following English sentence into French: 'The quick brown fox jumps over the lazy dog.'"

2

Creative Writing:

Task: Request the model to write a short story or piece of creative content.

Example Prompt: "Create an engaging story about a time traveler who discovers a hidden civilization in the distant future."

3

Coding Assistance:

Task: Get help with writing a code snippet.

Example Prompt: "Write a Python function that calculates the Fibonacci sequence up to the 10th term."

Few tasks and their sample prompt:

1

Debate or Opinion:

Task: Engage the model in a debate or solicit its opinion on a topic.

Example Prompt: "Argue for and against the use of genetically modified organisms in agriculture."

2

Data Analysis:

Task: Ask the model to perform data analysis or calculations.

Example Prompt: "Calculate the average revenue of our company for the past year based on the provided dataset."

3

Customer Support:

Task: Use the model as a virtual assistant for customer inquiries.

Example Prompt: "Assist the customer by providing instructions on how to troubleshoot a network connectivity issue."

Content:

Prompting strategies



Open ended prompts:

Open ended prompts provide no specific direction or constraints to the user. Instead, they encourage open exploration and creativity. Open-ended prompts can be useful for generating ideas, encouraging brainstorming, and promoting imaginative responses.

Example:

How do you think AI will progress in the coming decade?

Closed ended prompts:

They provide specific guidelines and instructions for the model to follow to perform specific task. These have constraints in order to ensure model doesn't do a lot of interpretation on its own and gives required answer.

Example: Write a 10 line summary on recent Russia-Ukraine war

Multi part prompting

Solving complex scenarios and handling multiple related situations at once.

prompt_2 = f""

Your task is to perform the following actions:

- 1 - Summarize the following text delimited by <> with 1 sentence in at most 30 words, and focusing mainly aspects that show customer intent to buy.
- 2 - Translate the summary into French.
- 3 - List each name in the French summary.
- 4 - Output a json object that contains the following keys: french_summary, num_names.

Use the following format:

Text: <text to summarize>

Summary: <summary>

Translation: <summary translation>

Names: <list of names in French summary>

Output JSON: <json with summary and num_names>

Text: <{text}>

...

Creative prompts:

These prompts allow the model to explore creativity with words and come up with different tones and scenarios. In models like GPT 3.5, even the temperature can be reduced to get more creative responses.

Example:"Create a catchy Instagram caption for a travel photo in Paris"

Factual prompts:

These prompts are constrained to deliver only factual information from the additional context given and hard check is done to ensure the model is not hallucinating.
For example: Who is the President of USA?

Content:

Prompting strategies



Zero shot prompting

Zero shot prompting techniques are instructions given to the model without any explanation with examples. It assumes the model understands the tasks clearly.

'Classify the text into neutral, negative or positive. Text: I think the vacation is okay.Sentiment:'

When zero-shot doesn't work, it's recommended to provide demonstrations or examples in the prompt which leads to few-shot prompting.

Few shot prompting

Few-shot prompting can be used as a technique to enable in-context learning where we provide demonstrations in the prompt to steer the model to better performance. In cases where few shot prompting doesn't work, one can go ahead with much advanced techniques like multi shot prompting and chain of thought prompting.

Few shot prompting

prompt = f"""

Summarize the text delimited by triple backticks
into a single sentence.

```{text}```

For example :

Input email:

"Thank you for connecting on call, we would like to go ahead and buy the product, please share the quotation."

Summary:

"Interested in product and asking quotation."

....

# Giving context:

System message

Assistant is an intelligent chatbot designed to help users answer their tax related questions.

Instructions

- Only answer questions related to taxes.
- If you're unsure of an answer, you can say "I don't know" or "I'm not sure" and recommend users go to the IRS website for more information.

User - "When do I need to file my taxes by?"

Assistant -

# Asking output in a given format:

```
prompt = f"""\n
```

Generate a list of top keywords from the given email text delimited by triple backticks.

Provide them in JSON format with the following keys:

email\_id, email\_content,list of top keywords.

```
```{text}```\n....\n....
```

Break down complex problems:

System prompt:

You're an AI assistant that helps people find information.

User prompt:

You'll read a paragraph, and then issue queries to a search engine in order to fact-check it.

PARAGRAPH

John Smith is married to Lucy Smith. They have five kids, and he works as a software engineer at Microsoft. What search queries should I do to fact-check this?

Now you'll extract factual claims, and issue queries to fact-check them. When issuing a query, use the function
SEARCH("query")

FACTUAL CLAIMS

Role prompting

prompt_1 = f"""You are a marketing expert. Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

"""

prompt_2 = f"""You are a communications specialist. Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

"""

prompt_3 = f"""You are a customer service representative. Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

"""

Multi step prompting

1

This technique breaks down an end goal into a series of small steps. The model has to go through each step before giving the final output. These steps make things easier for the model and increase the chances of success.

2

As the task needs to be done in a specific order, it becomes helpful in directing the LLM.

3

A multi-step prompt breaks down a complex task into smaller, manageable steps that the model has to take to reach the desired outcome.

Multi step prompting

1

This technique breaks down an end goal into a series of small steps. The model has to go through each step before giving the final output. These steps make things easier for the model and increase the chances of success.

2

As the task needs to be done in a specific order, it becomes helpful in directing the LLM.

3

A multi-step prompt breaks down a complex task into smaller, manageable steps that the model has to take to reach the desired outcome.

Multi step prompting benefits

- 1 **Enhanced Accuracy and Relevance:** By dividing a task into smaller steps, LLMs can handle detailed and complex requests more accurately.
- 2 **Improved User Interaction:** Multi-step prompts allow for a more interactive and engaging experience.
- 3 **Complex Problem Solving:** Some tasks, due to their complexity, cannot be tackled in a single step without losing detail or accuracy.
- 4 **Learning and Adaptation:** Through multi-step interactions, LLMs can learn from the user's feedback, making real-time adjustments to better suit their preferences.

Steps in multi step prompting

- 1 Break down the problem
- 2 Give clear instructions
- 3 Stay consistent with the prompt steps

Example

- 1 Step 1: Write a short story about a character who discovers they have the ability to time travel.
- 2 Step 2: Describe how the character uses this ability to change events in their past.
- 3 Step 3: Show the consequences of these changes on the character's present and future.
- 4 Step 4: Conclude the story with a reflection on the character's journey and the lessons they've learned about the nature of time and destiny.

Tree of thought prompting

- 1 Tree of Thoughts prompting – that solves problems by explicitly decomposing them into a series of thoughts, or intermediate steps.
- 2 Similar to chain of thoughts prompting, tree of thoughts prompting generates a solution that is simply a sequence of individual thoughts.
- 3 However, this approach goes further by allowing multiple reasoning path to be considered at once – forming a tree of potential thoughts or reasoning paths – and exploring this entire solution space via LLM-powered self-evaluation.
- 4 With tree of thoughts prompting, the LLM can deliberately plan its solution, test various intermediate reasoning paths, and even perform backtracking, allowing the model to explore the solution space and eventually generate the correct output.

Content:

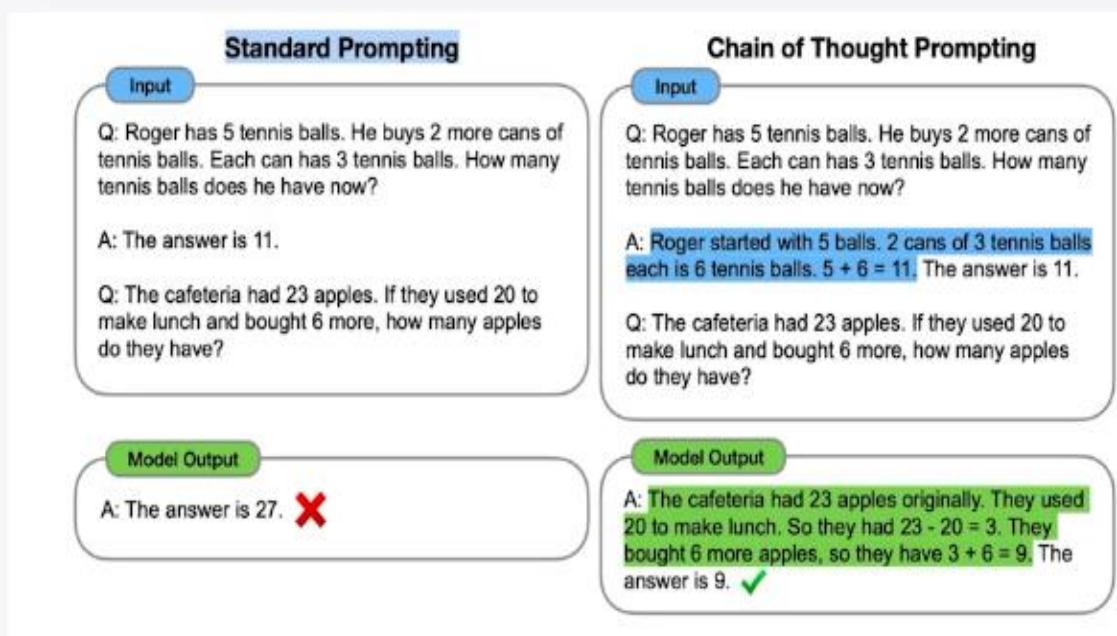
1. Prompt engineering-
Chain of thought
prompting



Chain of thought prompting

- 1 This technique allows LLMs to explain their reasoning hence give out much better and detailed results.
- 2 The CoT approach guides the LLM to unfold its reasoning, often leading to more accurate and interpretable results.
- 3 The main idea is that by giving few shot examples to LLMs by explaining the reasoning behind the result, LLMs would also come up with a reasoning for their answers and thus give out correct answers.

Chain of thought prompting



Regular Prompting vs CoT (Wei et al.)

Chain of thought prompting

1 This technique is especially helpful when solving mathematical problems where there is 1 correct answer and the reasoning is really necessary.

2 This has been shown to be effective in improving results on tasks like arithmetic, commonsense, and symbolic reasoning tasks.

3 It has been observed that with Chain of thought prompting as compared to the standard prompts, it shows 57% solve rate accuracy as compared to 18% with standard prompts on mathematical problems.

Chain of thought prompting

1

The user can add the instruction: 'Explain your answer step by step' in order to allow LLM to also perform the reasoning along with the actual solution,

2

Chain-of-thought prompting asks an LLM to mimic this process of decomposing a problem and working through it step by step -- essentially, asking the model to "think out loud," rather than simply providing a direct solution.

3

It has also been observed that this prompting technique delivers better results than standard prompting in most cases where it was used with Larger Language models and does not work well with smaller models.

Benefits of Chain of thought prompting

- 1 Better accuracy and precision
- 2 With logical reasoning targets the right source of information
- 3 Reduces likelihood of missteps and oversights and easier to identify errors.

Chain of thought vs prompt chaining:

Chain of thought:

- Chain-of-thought prompting asks the model to describe the intermediate steps used to reason its way to a final answer within one response
- Useful for planning and reasoning, such as math problems and logic puzzles, where explaining the thought process is essential to fully understanding the solution.

Prompt chaining:

- Prompt chaining involves an iterative sequence of prompts and responses, in which each subsequent prompt is formulated based on the model's output in response to the previous one.
- Useful technique for more creative, exploratory tasks that involve gradual refinement, such as generating detailed narratives and brainstorming ideas.

Automatic chain of thought prompting

Automatically generates the intermediate reasoning steps by utilizing a database of diverse questions grouped into clusters.

Auto-CoT goes through two main stages:

- 1 Question Clustering: First, they partition questions of a given dataset into a few clusters. So, if people asked the computer program a bunch of questions about baking, like "How do I mix the ingredients?" and "What temperature should I bake a pie at?" these would go into different groups.
- 2 Demonstration Sampling: Once they have these question groups, they pick one question from each group and use Zero-Shot CoT prompt (basically the "Let's think step by step" prompt). This way, the computer program generates clear and straightforward instructions on auto-pilot.

Content:

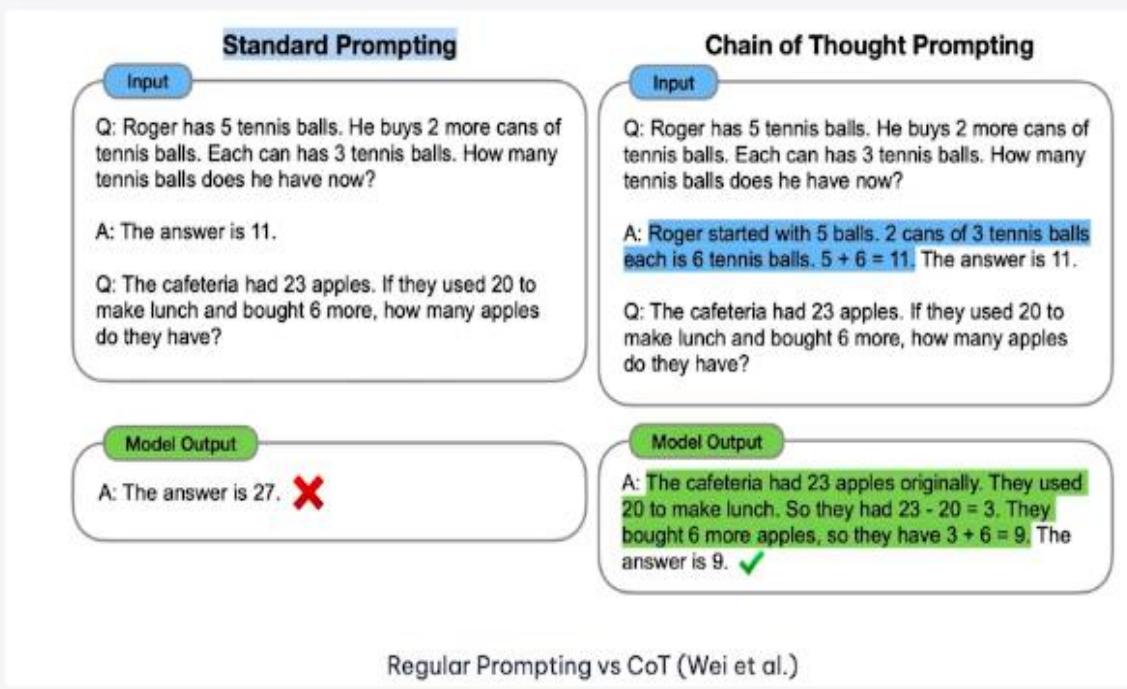
1. Prompt engineering-
Chain of thought
prompting



Chain of thought prompting

- 1 This technique allows LLMs to explain their reasoning hence give out much better and detailed results.
- 2 The CoT approach guides the LLM to unfold its reasoning, often leading to more accurate and interpretable results.
- 3 The main idea is that by giving few shot examples to LLMs by explaining the reasoning behind the result, LLMs would also come up with a reasoning for their answers and thus give out correct answers.

Chain of thought prompting



Chain of thought prompting



This technique is especially helpful when solving mathematical problems where there is 1 correct answer and the reasoning is really necessary.



This has been shown to be effective in improving results on tasks like arithmetic, commonsense, and symbolic reasoning tasks.



It has been observed that with Chain of thought prompting as compared to the standard prompts, it shows 57% solve rate accuracy as compared to 18% with standard prompts on mathematical problems.

Chain of thought prompting

- 1 The user can add the instruction: 'Explain your answer step by step' in order to allow LLM to also perform the reasoning along with the actual solution,
- 2 Chain-of-thought prompting asks an LLM to mimic this process of decomposing a problem and working through it step by step -- essentially, asking the model to "think out loud," rather than simply providing a direct solution.
- 3 It has also been observed that this prompting technique delivers better results than standard prompting in most cases where it was used with Larger Language models and does not work well with smaller models.

Chain of thought prompting

- 1 The user can add the instruction: 'Explain your answer step by step' in order to allow LLM to also perform the reasoning along with the actual solution,
- 2 Chain-of-thought prompting asks an LLM to mimic this process of decomposing a problem and working through it step by step -- essentially, asking the model to "think out loud," rather than simply providing a direct solution.
- 3 It has also been observed that this prompting technique delivers better results than standard prompting in most cases where it was used with Larger Language models and does not work well with smaller models.

Benefits of Chain of thought prompting

- 1 Better accuracy and precision
- 2 With logical reasoning targets the right source of information
- 3 Reduces likelihood of missteps and oversights and easier to identify errors.

Chain of thought vs prompt chaining:

Chain of thought:

- Chain-of-thought prompting asks the model to describe the intermediate steps used to reason its way to a final answer within one response
- Useful for planning and reasoning, such as math problems and logic puzzles, where explaining the thought process is essential to fully understanding the solution.

Prompt chaining:

- Prompt chaining involves an iterative sequence of prompts and responses, in which each subsequent prompt is formulated based on the model's output in response to the previous one.
- Useful technique for more creative, exploratory tasks that involve gradual refinement, such as generating detailed narratives and brainstorming ideas.

Automatic chain of thought prompting

Automatically generates the intermediate reasoning steps by utilizing a database of diverse questions grouped into clusters.

Auto-CoT goes through two main stages:

- 1 Question Clustering: First, they partition questions of a given dataset into a few clusters. So, if people asked the computer program a bunch of questions about baking, like "How do I mix the ingredients?" and "What temperature should I bake a pie at?" these would go into different groups.
- 2 Demonstration Sampling: Once they have these question groups, they pick one question from each group and use Zero-Shot CoT prompt (basically the "Let's think step by step" prompt). This way, the computer program generates clear and straightforward instructions on auto-pilot.

Content:

Bias in prompts



Bias in prompts

- 1 Prompts should be designed such that from the user's end we do not introduce bias in the prompts
- 2 Prompts should be structured neutrally in order to achieve better results.
- 3 Foundational language models like ChatGPT are trained on massive amount of textual data extracted from the internet, including books, articles, code repositories, and web text.

Bias in prompts

While this actual dataset might not be known we know based on our understanding of the real world, these biases could be reflected in the model's learning:

- 1 Social and cultural biases: Language itself is capable of encoding bias in terms of gender, race, ethnicity, religion or other social categories. Thus sometimes the output could come out as abusive language or negative stereotypes. Famous examples could be mentioning nurse as a female gender.
- 2 History biases: Textual data often reflects historical biases that may no longer be considered acceptable
- 3 Algorithmic bias: By prioritizing some types of information over others, algorithms used for the processing and selection of training data can create biases.

Examples of bias in prompts

For example giving a prompt like: Explain why robots are helpful in medical surgery ? would bias the model to have an understanding that robots are definitely helpful. Instead prompts like: Are robots helpful in medical surgery ? If so, explain why? Can give better results as it would give out few nuances where it could not be helpful as well.

Examples of bias in prompts

Bias related to gender is very common and a prompt such as 'Create a job description for a CEO' could return biased answer focusing on a male candidate or on certain race. Thus a better prompt could be 'Create a job description for a data scientist, ensuring the language is unbiased and inclusive.'

It is better to ask for a neutral and inclusive response in order to get detailed and diverse response.

Bias in prompts

- 1 Over the past years through multiple researches it has been observed that models such as Google translate, Facial recognition systems, resume shortlisting systems show biases in terms of culture , gender and race.
- 2 With LLMs multiple strategies can be employed while evaluating the responses in order to get better debiased responses both with prompt engineering, model fine tuning and smart evaluation strategies.

Bias in prompts

1

When using Chat GPT, it is important to consider bias in the prompt's wording, as well as the impact that may have in the resulting output. Trying different prompts is one way to obtain additional perspectives on a topic.

2

Even if the wording of the prompt is neutral, attention should be paid to the output generated by ChatGPT. A very slight difference in the prompt has deep ramifications in the generative output.

Debiasing strategies:

- 1 These are strategies to ensure model responses are not skewed towards certain biases.
- 2 These can include debiasing the prompts to get more coverage and ensure neutral inputs, or even finetuning the models with enough training data, and data augmentation techniques to mitigate bias.
- 3 Also while evaluating the prompt responses, special checks can be done to check for biased responses.
- 4 Prompt review and testing and further iterations are needed to ensure model maintains a neutral stand in such cases.

Debiasing Prompting strategies:

Exemplar prompting:

By giving few shot examples trying to neutralize the input data response, we can get more reliable responses.

- 1 This also includes ordering and distribution of the examples.
For example if performing sentiment analysis, how many examples from each class is available.
- 2 Also ensuring no bias in ordering and getting randomly ordered examples with enough coverage can help neutralize the inputs.

Debiasing Prompting strategies:

Instruction debiasing:

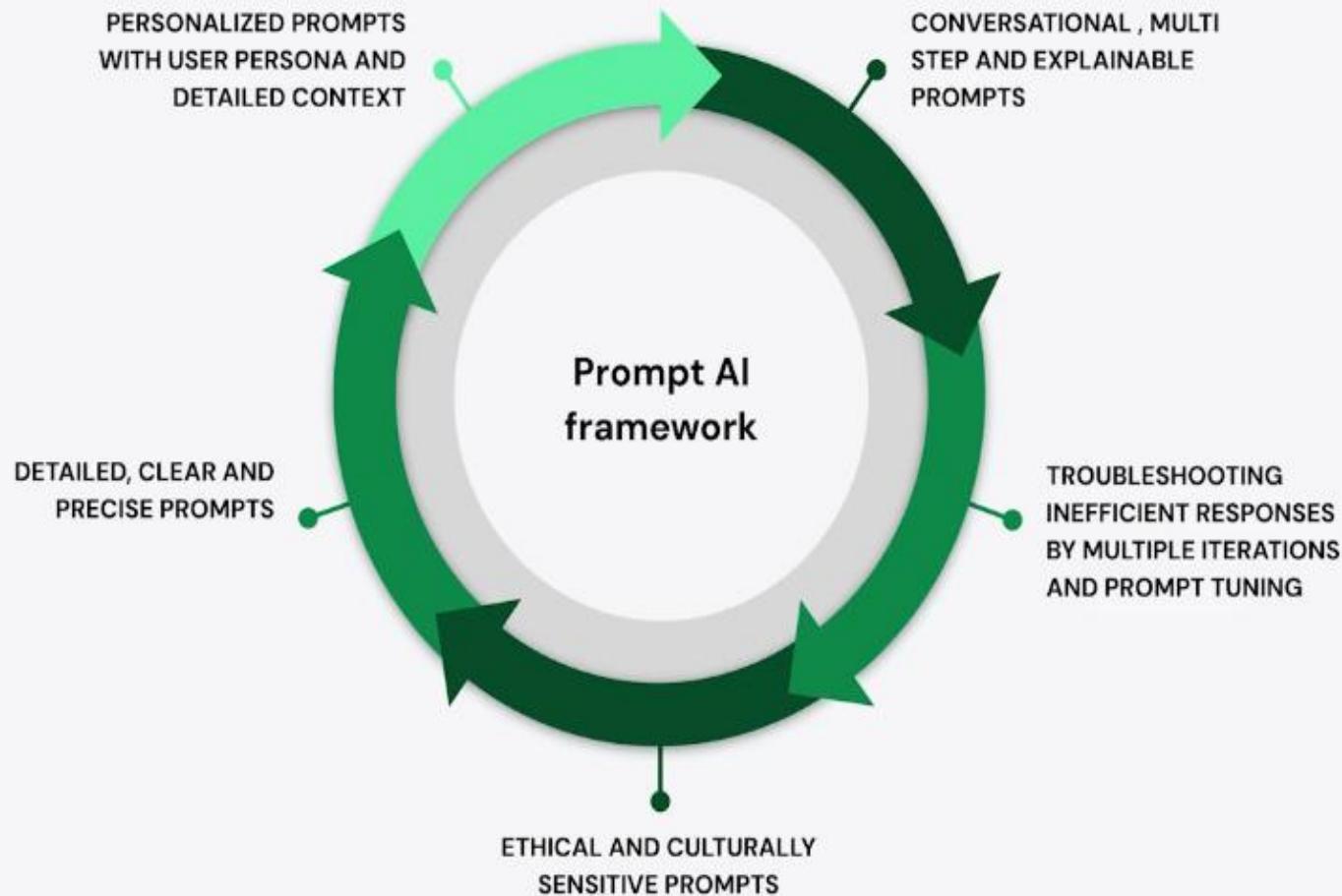
Giving specific instructions to eliminate the bias, such as when asking for job description, we specifically ask for inclusion of all caste, gender and race and avoid any pre existing stereotypes.

Content:

Responsible prompt
engineering



Responsible prompting Framework



Responsible prompt engineering

- 1 Prompt engineering is going to be a very useful skill as it guides the advanced foundational models to interact with humans in a conversational manner. This is really helpful in creating helpful chatbots, automating customer support and generating creative content.
- 2 However, if prompted the wrong way, it is not hard to generate harmful content from these models as well. So we need to be really careful with the prompts.

Responsible prompt engineering

- 1 As AI's capabilities continue to evolve, so too must the ethical frameworks that govern them. This intricate relationship between technology and ethics is at the heart of modern AI development.
- 2 Ethical prompt engineering is about anticipating and mitigating the risks associated with AI decision-making and interactions
- 3 It's about creating AI systems that are not only smart but also wise – capable of distinguishing right from wrong, sensitive to the diverse needs of users, and adaptable to the evolving landscape of ethical norms in technology.

Ethical challenges with AI:

- 1 Toxic content generation: As these AI models are trained on lots of textual data gathered from the internet , it can also include harmful data such as cuss words or slangs. Thus it becomes a challenge to ensure that while building applications like conversational bots, they do not respond back in any harmful language to the customers.
- 2 Few techniques used are using guardrail models if using foundational models such as ChatGPT to filter out such harmful content before passing on to customers.
- 3 If using LLMs which can be fine tuned, it is suggested to train them on diverse content of data, and even use techniques like RLHF to make the model distinguish better between harmful and toxic content.

Ethical challenges with AI:

- 1 LLM hallucinations:
LLM hallucinations are quite common as these models contain wide variety of textual context understanding without much details on distinguishing between facts and creative content.
- 2 It becomes a challenge to ensure these models don't return out fake information to customers thus guiding them in wrong direction.

LLM hallucinations

In order to cater to LLM hallucinations following techniques can be tried out:

- 1 Model parameters: Model parameters can be carefully chosen such that we restrict model's creative freedom and ensure its more deterministic. For these parameters such as temperature, top_p etc can be made more stricter.
- 2 Careful prompt designing: Prompts can be designed such that models can be instructed to stick to facts and not get creative when responses are not known.
- 3 RAG and source verification: Asking the model to state the source and giving additional knowledge source via RAG pipelines.

Ethical challenges with AI:

- 1 Legal challenges:
This involves data privacy related issues as well usage of copyright data in model training.
- 2 This also involves LLM responses generated by being trained on intellectual property related data and how does legal rules apply when such content is generated.
- 3 While deploying, it must be ensured that the data privacy is ensured and model does not output any personal information to customers.

Industrial applications:

Few industries where ensuring ethical prompt engineering can shape interactions with AI and is really needed:

- 1 Healthcare: Careful consideration is needed to prevent LLM hallucinations, securing customer medical records privacy and careful handling of sensitive patient data.
- 2 Public governance: Usage of AI in delivering the right government services with complete legal bindings.
- 3 Financial services: Providing fair, transparent financial recommendations while ensuring data privacy.
- 4 Education: AI systems can adapt to personalized recommendations with complete inclusion and proving unbiased educational responses.

RAG- Retrieval Augmented Generation

- 1 Enhanced Information Retrieval: RAG combines the strengths of retrieval-based models and generative models. It first uses a retrieval component to find relevant documents or passages from a large corpus based on a given query.
- 2 Contextual Response Generation: After retrieving relevant information, the generative component of RAG uses this context to produce more accurate and contextually appropriate responses.
- 3 Scalability and Flexibility: RAG is highly scalable and can be adapted to various domains by simply changing the corpus from which it retrieves information.

RLHF

RLHF: Reinforcement learning from human feedback

RLHF allows training by inclusion of human feedback using a small set of labelled dataset thus improving the context of the model in a better way.

RLHF techniques:

- 1** **Reward modeling:** Here the model generates several possible outputs or actions, and human evaluators rank or rate these outputs based on their quality. The model then learns to predict these human-provided rewards and adjusts its behavior to maximize the predicted rewards.

- 2** **Proximal policy optimization (PPO)** is an iterative algorithm that updates the language model's policy to maximize the expected reward. The core idea of PPO is to take actions that improve the policy while ensuring the changes are not too drastic from the previous policy.

Responsible prompt engineering

- 1 The best way forward is to ensure prompts are catering to mitigating biases, and contain user's persona and complete context such that it ensures complete inclusion.
- 2 Providing enough examples, giving additional context, thorough evaluation and iteration can help with mitigating some of the challenges seen above.
- 3 Using RAG based systems & RLHF based training techniques can be used for minimizing hallucinations.
- 4 Careful steps can be taken towards ensuring data privacy, ensuring safeguards such that end users are never exposed to outputs that contain sensitive data, contain abusive or antisocial messages.

Content:

1. Prompt engineering tools



Prompt engineering

- 1 Prompt engineering is a heavy and challenging task which involves multiple iterations and checks in order to generalize and scale it for multiple customers.
- 2 It involves a number of processes, including data preparation, the crafting of custom prompts, the execution of these prompts through the LLM API, and the refinement of the generated content.

Prompt engineering

1

Challenges:

1. Design and development
2. Evaluation and refinement
3. Optimization and production

2

In order to respond to these challenges 'Prompt flow' has been developed, which expedites and simplifies the development, evaluation, and continuous integration and deployment of prompt engineering.

3

These tools guide users through the journey from initial ideation and experimentation, culminating in the creation of production-ready applications powered by large language models (LLMs).

Prompt engineering tools:

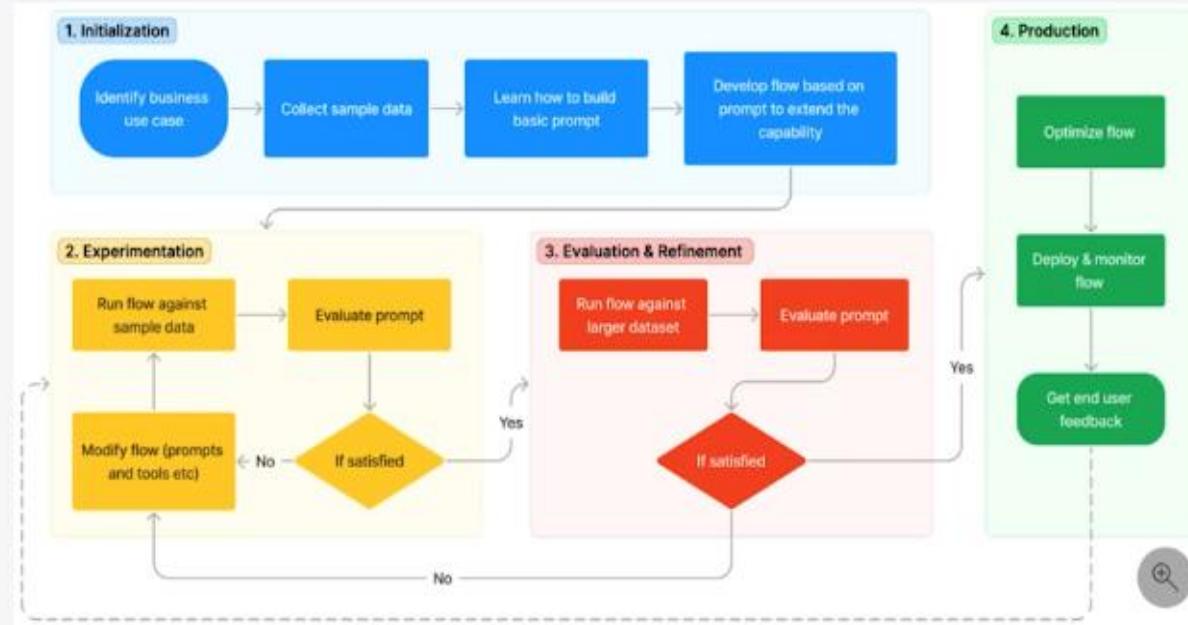
- 1 Azure prompt flow
Google AI studio
OpenAI playground
- 2 TensorOps LLM studio
Langsmith by Langchain
- 3 PromptAppGPT
Promptify
Jupyter Notebooks

Azure prompt flow:

Azure Machine Learning prompt flow offers:

- 1 Create executable flows that link LLMs, prompts, and Python tools through a visualized graph.
- 2 Debug, share, and iterate flows with ease through team collaboration. Create prompt variants and evaluate their performance through large-scale testing.
- 3 Deploy a real-time endpoint that unlocks the full power of LLMs for your application.

Azure prompt flow:



TensorOps LLM studio

Powered by TensorOps, LLMstudio is an evolving environment where users can experiment, modify, and optimize their interactions with advanced language models.

Benefits include:

- 1 Streamlined Prompt Engineering: Simplify and enhance prompt design process.
- 2 Execution History: Keep a detailed log of past executions, track progress, and make iterative improvements effortlessly.
- 3 Effortless Data Export: Share and exporting data to shareable CSV files.

This can be installed using '*pip install llmstudio*'

Langsmith:

LangSmith is a platform designed to simplify debugging, testing, evaluating, and monitoring large language model (LLM) applications

Key Features of LangSmith include:

- 1 Debugging:** It offers full visibility into the sequence of model inputs and outputs, enabling quick identification and resolution of errors.
- 2 Testing & Evaluating:** It provides a simple way to create and manage test datasets. Also LangSmith integrates with evaluation modules, employing both heuristic logic and LLMs themselves to evaluate the correctness of an answer.
- 3 Monitoring:** It offers tools to monitor system-level performance (such as latency and cost), and track user interactions.
- 4 Unified Platform:** Lang Smith serves as an integrated hub for all stages of LLM application development, streamlining the development process.

Furthermore, LangSmith supports data export in formats compatible with OpenAI evaluations and analytics engines, promoting easy fine-tuning and analysis of models. It's currently in **closed beta**.

Google AI studio:

- 1 Google AI studio leverages google cloud capabilities to tune and augment Gemini, with security compliances and deploy AI powered apps easily.
- 2 They also provide access to Gemini models , along with google developer ecosystem, including tools like colab, haystack, langchain, llama index to build scalable AI powered systems such as conversational bots.

Open AI playground

- 1 Open AI playground contains elements which enables users to configure models to their specific needs. Key aspects like Mode, System Prompts, and Model selection introduce users to advanced customization options.
- 2 It supports the application of system prompts, allowing users to tailor AI responses to specific scenarios or styles, enhancing the AI's versatility.

PromptAppGPT

- 1** PromptAppGPT is a low-code prompt-based rapid app development framework. PromptAppGPT contains features such as low-code prompt-based development, GPT text generation, DALLE image generation, online prompt editor+compiler+runner.
- 2** PromptAppGPT significantly lowers the barrier to GPT application development, allowing anyone to develop AutoGPT-like applications with a few lines of low code.

Promptify:

- 1 Promptify specially focuses on helping with generating prompts for various use cases such as creative content generation, marketing logo design, story writing and many more.
- 2 Similar to promptify there are various other alternate tools such as Prompt Box, promptly, Promptpal and many more.

Jupyter notebooks:

- 1 Jupyter notebooks help python developers with giving them flexibility to connect with various data sources, integrate seamlessly with existing codebases, and enable real-time code execution with immediate visual feedback.
- 2 This interactive environment encourages exploratory analysis and iterative coding, making it an ideal platform for developing, documenting, and executing data-intensive workflows.

Content:

Fine tuning language
models



Large language models

1

LLMs are foundational models, trained on massive datasets of text, perform a wide range of tasks, including text generation, translation, summarization, and question answering.

2

LLMs today are widely used for such text generation applications with substantial amount of prompt engineering, evaluation and heavy deployment steps.

3

While these LLMs are good at text generation, in order to get better accuracy at certain domains, fine tuning techniques are applied.

4

Fine-tuning allows users to adapt pre-trained LLMs to more specialized tasks. By fine-tuning a model on a small dataset of task-specific data, you can improve its performance on that task while preserving its general language knowledge.

Fine tuning LLMs

The need to fine tuning LLMs arise if :

- 1 Customization is required in certain language/ domains. For example legal documents can be used to fine tune and customize LLMs.
- 2 Building in house models to ensure data privacy
- 3 Building domain specific inhouse LLMs with limited data, as it leverages the knowledge learned during pre-training, saving substantial time and computational resources that would otherwise be required to train a model from scratch.

LLM Fine tuning techniques

Supervised fine tuning:

The model is trained on task specific labelled dataset, where model learns to adjust its parameters to predict these labels as accurately as possible. This is the most common fine tuning technique, which can improve model's performance even with less labelled data.

LLM Fine tuning techniques

Supervised fine tuning:

- 1** **Hyper parameter fine tuning:** Adjusting the hyper parameters such as the learning rate, batch size, and the number of epochs. Optimal hyperparameters can significantly enhance the model's performance on the specific task.
- 2** **Transfer learning:** Models pre trained on larger corpus are fine tuned on smaller labelled data for task specific requirements. It leads to reduced amount of data and training time required and often leads to superior performance

LLM Fine tuning techniques

Supervised fine tuning:

- 1 **Multi task learning:** Here the model is fine tuned on multiple tasks simultaneously with joint losses in order to learn the context in a better way and the model can develop a more robust and generalized understanding of the data.
- 2 **Task specific fine tuning:** Task-specific fine-tuning is closely related to transfer learning, but transfer learning is more about leveraging the general features learned by the model, whereas task-specific fine-tuning is about adapting the model to the specific requirements of the new task.

LLM Fine tuning techniques

RLHF: Reinforcement learning from human feedback

RLHF allows training by inclusion of human feedback using a small set of labelled dataset thus improving the context of the model in a better way.

RLHF techniques:

1

Reward modeling: Here the model generates several possible outputs or actions, and human evaluators rank or rate these outputs based on their quality. The model then learns to predict these human-provided rewards and adjusts its behavior to maximize the predicted rewards.

2

Proximal policy optimization (PPO) is an iterative algorithm that updates the language model's policy to maximize the expected reward. The core idea of PPO is to take actions that improve the policy while ensuring the changes are not too drastic from the previous policy.

LLM Fine tuning techniques

RLHF techniques:

- 1 **Comparative ranking:** Here the model learns the relative ranking given by humans for multiple outputs.
- 2 **Preference learning:** It focuses on training models to learn from human feedback in the form of preferences between states, actions, or trajectories. In this approach, the model generates multiple outputs, and human evaluators indicate their preference between pairs of outputs.
- 3 **Parameter-efficient fine-tuning (PEFT)** is a technique used to improve the performance of pre-trained LLMs on specific downstream tasks while minimizing the number of trainable parameters.

Fine tuning applications

Summarization

Chatbots

Sentiment
analysis

Content:

Evaluating prompt outputs



Evaluating prompt outputs

- 1 Evaluating prompt responses is a crucial step as it helps us understand the requirements for iteration and better user experience.
- 2 Robust evaluation of models ensures that it not only adheres to technical specifications but also resonates with user expectations and proves its worth in practical scenarios.
- 3 Evaluation not only helps in improving model performance, but also builds trust in AI solutions and stress tested prompts.

Evaluating LMs traditionally

- 1 Language modeling is the task of predicting the next word or character in a document
- 2 Evaluation can be divided into Intrinsic and Extrinsic evaluation.
- 3 Intrinsic evaluation captures how well the model captures what it is supposed to capture, like probabilities.
- 4 Extrinsic evaluation (or task-based evaluation) captures how useful the model is in a particular task

Evaluating LMs traditionally

- 1 Language modeling is the task of predicting the next word or character in a document
- 2 Evaluation can be divided into Intrinsic and Extrinsic evaluation.
- 3 Intrinsic evaluation captures how well the model captures what it is supposed to capture, like probabilities.
- 4 Extrinsic evaluation (or task-based evaluation) captures how useful the model is in a particular task

Traditional evaluation methods

If the LM is further used for classification, metrics like accuracy, F1, AUC are tracked.

Embeddings from LMs can also be plotted on TSNEs and visualized for better evaluation.

Human evaluation on criteria like relevance, fluency, and coherence is also considered

Other metrics like BLEU, ROUGE score have also been used for generation tasks

Evaluating prompt outputs

- 1 A good evaluation would involve outcomes correlated with human judgement, fast and automated evaluation, stress tested on diverse scenarios and clear display of metrics.
- 2 While traditional NLP methods are useful, they are limited as these limitations primarily arise from the metrics' inability to fully capture the nuanced understanding and generation of human language.
- 3 For example BLEU score focuses predominantly on the precision of word matches, often overlooking the context and semantics.

Evaluating prompt outputs

Type	Description	Example metrics
Diversity	How versatile the outputs are	ROUGE, Perplexity
User feedback	Quality of responses	Coherence, quality, relevance
Ground truth based metrics	Where we have golden set to compare	Accuracy, F1, Precision, Recall
Answer relevance/ correctness	Is the answer relevant to context and correct without hallucinations	Binary classification (Yes/No)
Toxicity analysis	Check if harmful content is generated?	Fairness scoring

Evaluating prompt outputs

Word and character based

- BLEU
- ROUGE
- WER
- Edit distance

Embedding based

- BERTScore
- Mover score

Language model
based

- NLI score
- BLEURT

LLM assisted

- GPT-score
- G eval
- Self check GPT

Traditional methods

- 1 **BLEU:**It evaluates candidate text based on how many ngrams in the generated text appear in reference text. This is based on precision calculation.
- 2 **ROUGE:** This is recall based. ROUGE-L measures the longest common subsequence (LCS) between a pair of sentences.
- 3 **WER:** Edit distance, number of insertions, deletions, transformations to match the string.

Non Traditional methods

- 1 **BertScore:** This is embedding based match, where cosine similarity helps in identifying how close the embeddings are for similar words and vice versa.
- 2 **Mover Score:** Based on Word mover distance , which allows one-to-one hard matching of words, MoverScore allows many-to-one matching as it uses soft/partial alignments.
- 3 **NLI score:** Generated using natural language inference models. This method can be very useful to ensure faithfulness for text-grounded generation tasks like text summarization.
- 4 **BLEURT:** Additional BERT based models are used where text collected is then augmented with techniques like word-replacements, back-translation, etc to form synthetic pairs and then trained on objectives like BLUE, ROGUE scores, back-translation probability, natural language inference, etc.

LLM assisted methods:

- 1 **GPT score:** It captures the ability of LLM in evaluating multiple aspects of generated text such as informativeness, relevancy, etc. by checking token probabilities for target response.
- 2 The approach assumes that the LLM will assign higher token probabilities for higher-quality generations and uses the conditional probability of generating the target text (hypothesis) as an evaluation metric.

LLM assisted methods:

1

G-eval: This directly performs evaluation by explicitly instructing the model to assign a score to generated text in the 0 to 5 range. This output is then multiplied by token probabilities for similar evaluation to GPT score.

2

SelfCheckGPT: This technique is combination of traditional and non traditional techniques, as it includes comparing the text with knowledge reference using RAG. This helps to keep hallucinations in check.

LLM assisted methods:

1

Limitations of these methods:

1. Application specific
2. Positional bias: Bias for ordering and positioning of data based on input prompts
3. Verbose bias: LLMs tend to prefer longer responses and if not prompted well, rarely delivers crisp responses.

2

Few measures taken to mitigate these involve few shot prompting with random positioning, along with hybrid evaluation such as integrating LLM-based assessments with human judgment or advanced non-traditional metrics can be highly effective.

Evaluation tools:

- 1 **OpenAI evals:** Using this framework you can evaluate the completions for instructions against a reference ground truth that you have defined.
- 2 **RAGAS:** It uses SOTA LLM-assisted methods to quantify the performance of RAG pipelines in multiple aspects like factuality, relevance, etc.

Content:

Common prompting
issues



Common prompting issues

1

Vague or ambiguous prompts:

Vague or ambiguous prompts are those that lack clarity, precision, or specificity. Vague or ambiguous prompts lead to confusion and unclear responses for several reasons.

2

Respondents might interpret the question differently from what was intended, or they might be unsure about the scope or context of the question.

3

It is suggested to be clear and concise in sending instructions and being specific about what needs to be returned as output.

Common prompting issues

1

Overly complex prompts:

Overly complex prompts can be difficult to comprehend, and cause confusions as to what actually needs to be done and in what order.

2

This can cause difficulty in comprehension, and leading to decreased quality of responses.

3

It is suggested to break the prompt into multiple components, while focusing on one task at a time with clear instructions. Also multiple iterations and sending examples can help.

Common prompting issues

1

Biased prompts:

Such prompts are skewed by the user itself directing the model to respond in a particular way thus skewing results.

2

These prompts might contain suggestive language, make assumptions, or present a specific viewpoint, causing respondents to feel compelled to respond in a manner that aligns with the prompt's bias.

3

It is suggested to have balanced prompts with multiple examples, not asking open ended questions which might lead the model to make certain assumptions

Common prompting issues

1

Ignoring the customer requirements:

Without understanding the customer requirements, it is difficult to give required user experience.

2

Understanding your audience's needs, preferences, and background is essential for crafting questions that resonate with them, and ultimately, for obtaining accurate and useful information.

Common prompting issues

1

Not providing enough context:

Providing sufficient context allows respondents to grasp the purpose of the prompt, the scope of the question, and any assumptions or limitations that may apply

2

Otherwise it leads to inconsistency in responses, inaccurate responses without models understanding the complete scope.

Content:

Future of prompt
engineering



Prompt Engineering

- 1 Prompt engineering bridges the communication gap between humans and Generative models.
- 2 It involves crafting clear and specific prompts or instructions to get the desired output from the model.
- 3 It is the process of formulating prompts or queries that guide the model's responses for a particular task or use case. Prompt engineering is essential to make GPT-3.5 produce accurate, relevant, and useful content.

Ongoing developments in prompt engineering

- 1 With advancements in LLMs and larger context being allowed for inputs, prompt engineering becomes an essential skill as most use cases can be solved by few shot prompting examples and RAG pipelines.
- 2 Prompt engineers along with writing good prompts are now also involved in data labeling, model optimization, knowledge extraction and content generation using AI.
- 3 There is also an increased demand across multiple industries.
The integration of prompt engineering in AI & data science fields is emerging.

Ongoing developments in prompt engineering

- 1 Enhanced contextual understanding allows for even complex prompts understanding and multi task problems to be solved easily by LLMs.
- 2 Adaptive prompting is an emerging trend where AI models are being developed to adjust their responses based on the user's input style and preferences
- 3 Multi modal prompt engineering: Multimodal models can process and respond to prompts that include a mix of text, images, and sometimes even audio inputs.
- 4 Real time prompt optimization: This technology assesses the prompt's clarity, potential for bias, and alignment with the desired outcome, offering suggestions for improvement.
- 5 Ethical prompting: There's a focus on crafting prompts that ensure fairness, transparency, and bias mitigation.

Content:

Conclusion and recap



Large language models

- 1 LLMs (Large language models) come under the umbrella of Generative AI, where they cater to delivering textual content using input instructions.
- 2 These are used to solve various problems on structured and unstructured data such as summarization, question answering, conversational bots etc.
- 3 Few examples of these LLMs include models such as ChatGPT, Gemini , LLama. These LLMs are foundational models, trained on a broad spectrum of unstructured and generalized data.

LLM use cases:

Text generation
- Summarization
- Content generation

Language translation

AI assistants
- Code assistants

Language insights
- Sentiment analysis
- NER

Prompt Engineering

1

Prompt engineering bridges the communication gap between humans and Generative models.

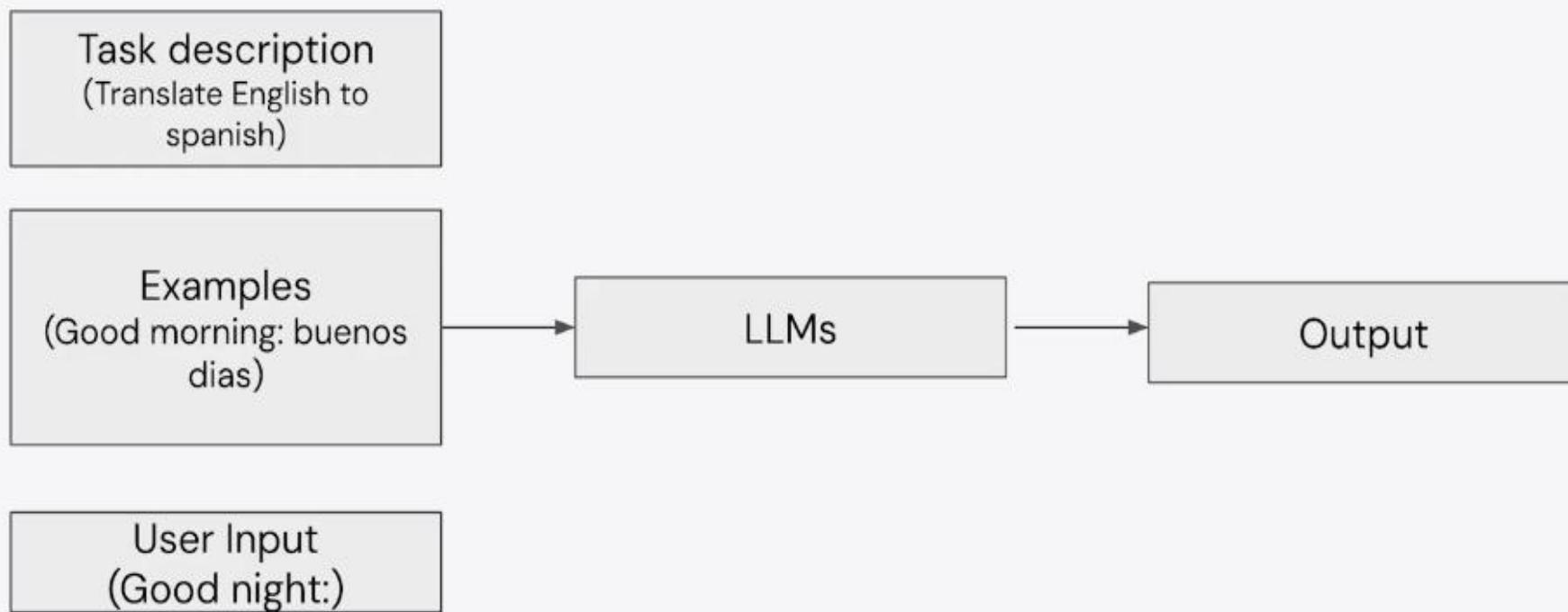
2

It involves crafting clear and specific prompts or instructions to get the desired output from the model.

3

It is the process of formulating prompts or queries that guide the model's responses for a particular task or use case. Prompt engineering is essential to make GPT-3.5 produce accurate, relevant, and useful content.

Prompt structure



Key considerations for Prompt engineering

- 1 **Clarity:** Ensuring that your prompt is clear and easy to understand.
- 2 **Contextual information:** When relevant, include background information or context in your prompts. This helps the model understand the task or question better.
- 3 **Specificity:** Provide specific details and context in your prompts, so the model has the necessary information to generate a relevant response.
- 4 **Desired format:** Specify the format or structure you want the answer in, whether it's a paragraph, list, table, or any other desired output format.

Few tips to follow:

1

Tone and style: If you have specific requirements for the tone, style, or level of formality in the response, make those expectations clear in your prompt.

2

Examples and demonstrations: You can provide examples or demonstrations in your prompt to illustrate the kind of response you're looking for.

3

Review and iterate: After receiving responses, review them to ensure they meet your expectations. If needed, iterate and refine your prompts to improve the model's performance.

4

Ethical considerations: Ensure that your prompts adhere to ethical guidelines and avoid generating harmful, offensive, or inappropriate content.

Few tasks and their sample prompt:

1

Question Answering:

Task: Ask the model to answer a specific question.

Example Prompt: "Can you explain the causes of climate change in a concise manner?"

2

Content Generation:

Task: Request the model to generate content on a particular topic.

Example Prompt: "Please provide an informative article about the benefits of renewable energy sources."

3

Summarization:

Task: Instruct the model to summarize a lengthy article or document.

Example Prompt: "Summarize the key points from the 10-page report on the impact of artificial intelligence on the job market."

Few shot prompting

prompt = f""

Summarize the text delimited by triple backticks
into a single sentence.

```{text}```

For example :

Input email:

"Thank you for connecting on call, we would like to go ahead and buy the product, please share the quotation."

Summary:

"Interested in product and asking quotation."

"""

# Break down complex problems:

System prompt:

You're an AI assistant that helps people find information.

User prompt:

You'll read a paragraph, and then issue queries to a search engine in order to fact-check it.

---

PARAGRAPH

John Smith is married to Lucy Smith. They have five kids, and he works as a software engineer at Microsoft. What search queries should I do to fact-check this?

---

Now you'll extract factual claims, and issue queries to fact-check them. When issuing a query, use the function  
SEARCH("query")

FACTUAL CLAIMS

# Chain of thought prompting

1

The user can add the instruction: 'Explain your answer step by step' in order to allow LLM to also perform the reasoning along with the actual solution,

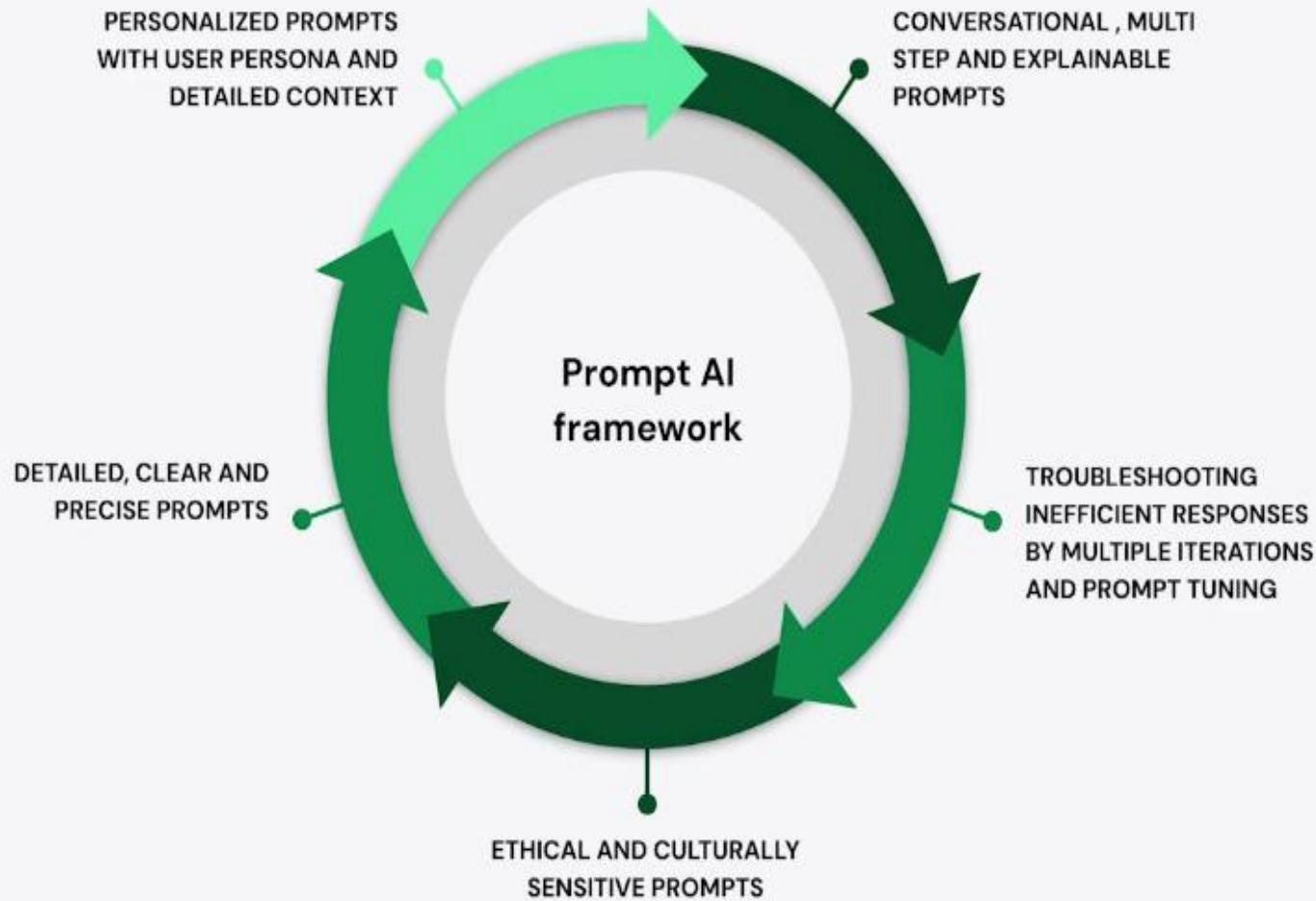
2

Chain-of-thought prompting asks an LLM to mimic this process of decomposing a problem and working through it step by step -- essentially, asking the model to "think out loud," rather than simply providing a direct solution.

3

It has also been observed that this prompting technique delivers better results than standard prompting in most cases where it was used with Larger Language models and does not work well with smaller models.

# Responsible prompting Framework



# Prompt engineering tools:

- 1 Azure prompt flow  
Google AI studio  
OpenAI playground
- 2 TensorOps LLM studio  
Langsmith by Langchain
- 3 PromptAppGPT  
Promptify  
Jupyter Notebooks

# Fine tuning LLMs

1

Fine-tuning adjusts the parameters of a pre-trained large language model to a specific task or domain.

2

Although LLMs are trained on vast language knowledge, they lack specialization in specific areas.

3

Fine-tuning addresses this limitation by allowing the model to learn from domain-specific data to make it more accurate and effective for targeted applications.

# Traditional evaluation methods

If the LM is further used for classification, metrics like accuracy, F1, AUC are tracked.

Embeddings from LMs can also be plotted on TSNEs and visualized for better evaluation.

Human evaluation on criteria like relevance, fluency, and coherence is also considered

Other metrics like BLEU, ROUGE score have also been used for generation tasks

# Future of Prompt engineering

1

As the demand for AI gains momentum, it is reasonable to look for potential opportunities to improve prompts for emerging use cases.

2

Prompt engineering will continue to merge linguistic nuance with technical understanding and will become an even more essential skill for interacting with AI systems.

3

The process will involve defining goals, crafting initial prompts, testing and analyzing responses, refining prompts through iteration, and implementing them in real-world applications.

# Write ChatGPT Prompt for Crafting Outline for a Presentation

## Prompt Template for Generating Outline

Assume you are a <Role/ Actor Name>,  
for a < Describe your Company/Product/Project >. You are expert in crafting PowerPoint Presentations  
Generate the Outline for a PowerPoint Presentation  
on <'title of presentation topic'> topic  
with the intended goal to <purpose/desired outcome>  
The target audience is < audience >  
The presentation should have < number of slide in presentation>  
slides  
The tone should be <tone e.g. formal, informal, humorous,  
professional >  
Please create one slide per section, along with the key points for each  
section.  
It needs to cover <key point 1>, < key point 2>, and  
< key point 3, ..>.  
<any other additional information number 1>  
<any other additional information number 2>

## Example Prompt for Outline of a Presentation

Assume you are a **Management Consultant**,  
for an **Insurance Company**. You are expert in crafting PowerPoint Presentations  
Generate the Outline for a PowerPoint Presentation  
on '**Exploring Opportunities for AI in Insurance Sector**' topic  
with the intended goal to **give the audience context into the current AI Landscape** so that they can understand how AI can be used in their work  
The target audience is **Middle and Junior staff in the Insurance company**  
The presentation should have **10 slides**  
The tone should be **formal**  
Please create one slide per section, along with the key points for each  
section.  
It needs to cover **ethical use of AI in Insurance**,  
**Key use cases of AI in Insurance** and other key topics relevant to use of  
**AI in Insurance**