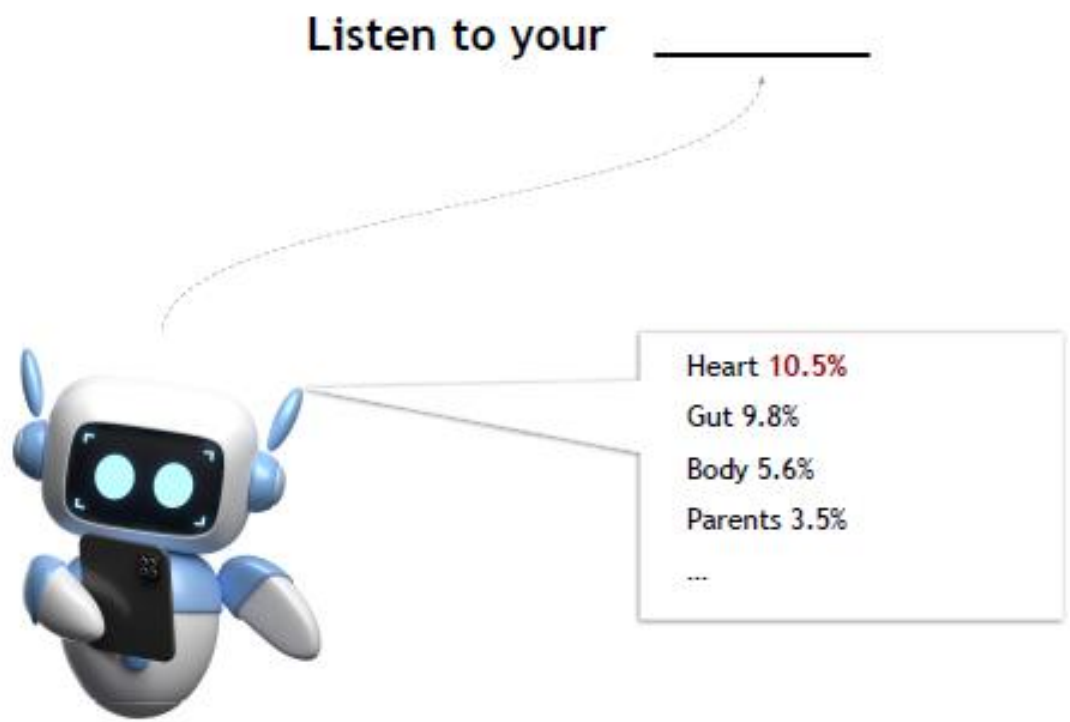
LLM and Prompt Engineering

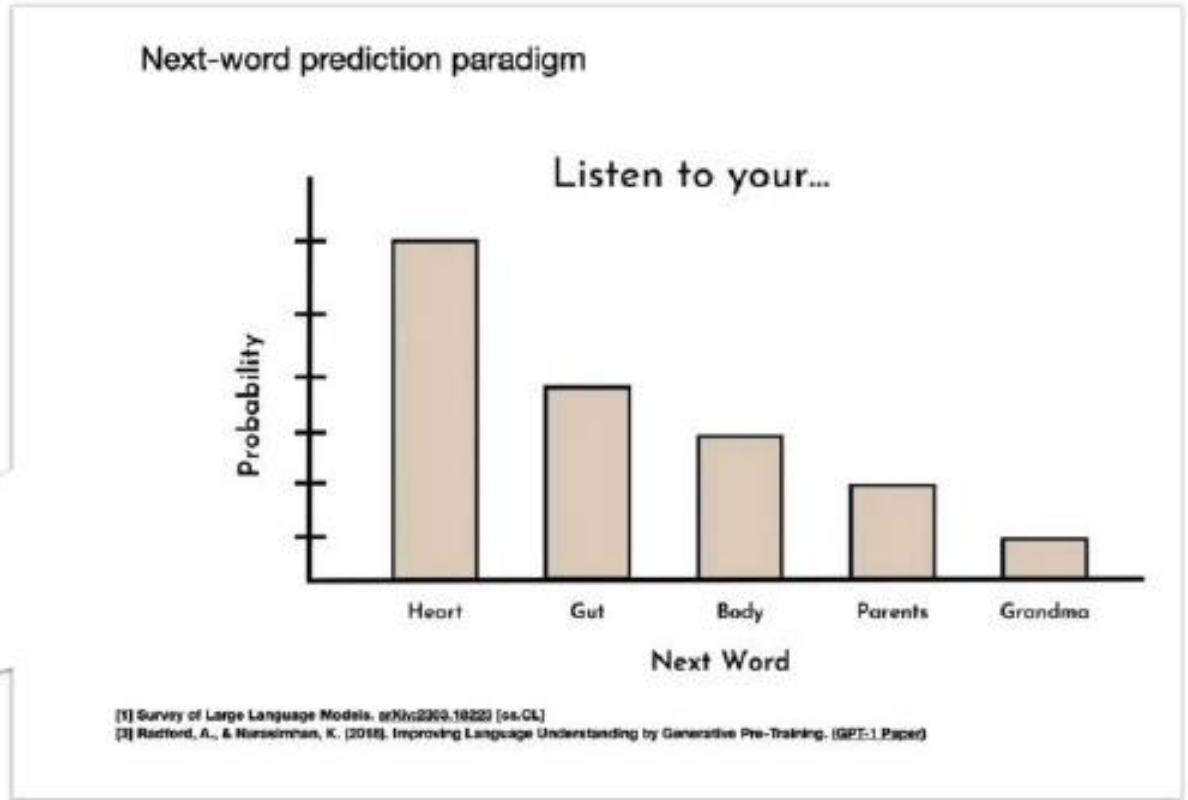
Redpanda-AI

What is a Language Model?

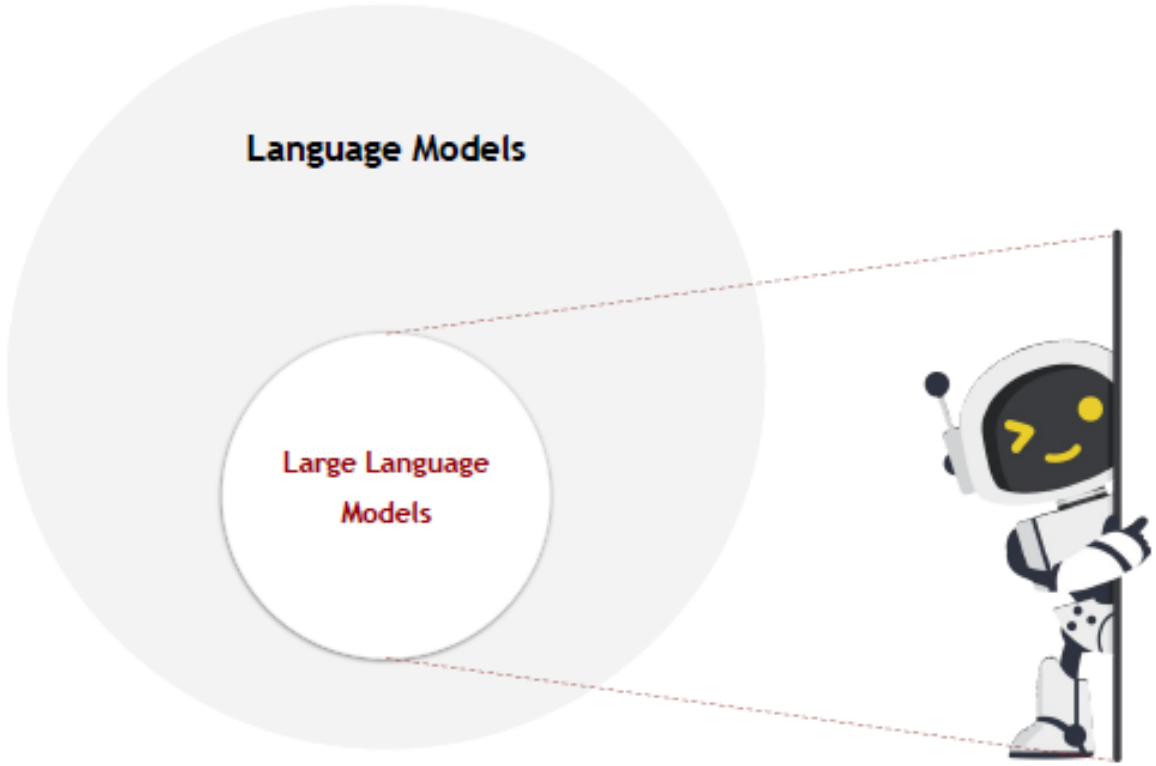
A language model is like a ***smart machine that helps you finish sentences***. It's trained to predict and complete sentences, making it a super useful tool when you're typing or texting



What is a Language Model?



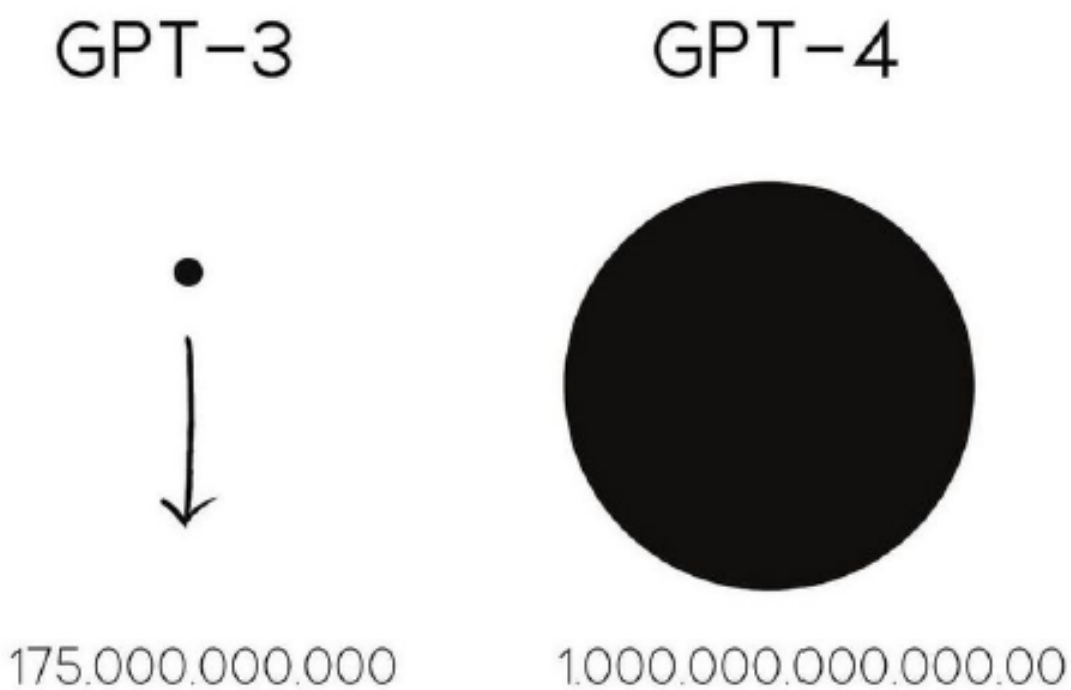
What are Large Language Models?



Models trained on large amounts of all type of publicly-available data that can perform a wide variety of tasks, including text/image/document summarization, generation, and categorization. These models can perform generative tasks like text/image/document generation and so there is some overlap between LLMs and Generative Al.

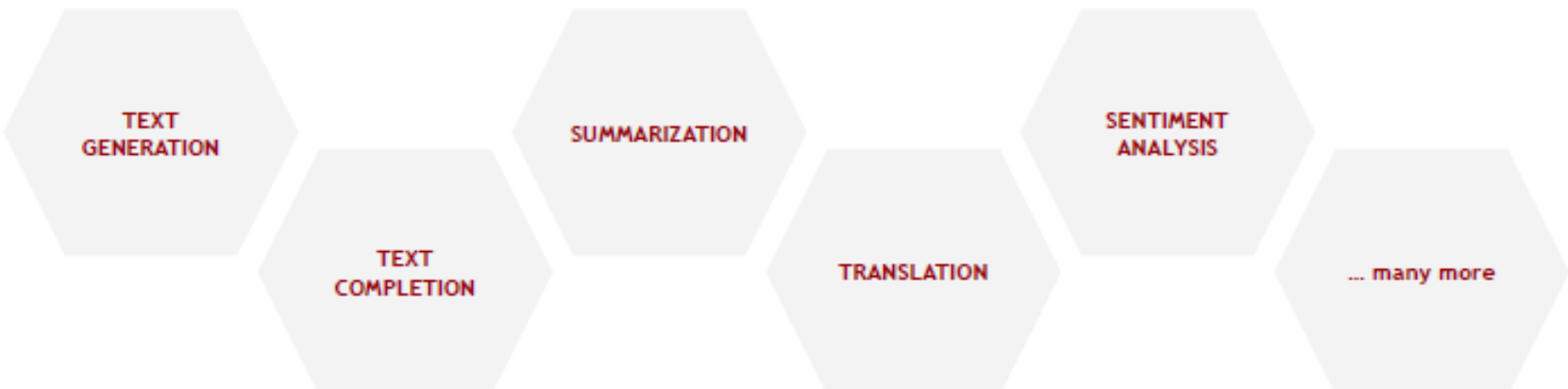
What is “**Large**” in Large Language Models?

- The term "**large**" refers to the size of the model in terms of its parameters and the volume of the training data
- Parameters are a part of the model that learns from historical training data
- They help in predicting the next word in a sequence
- One Trillion Parameters have been used in GPT 4



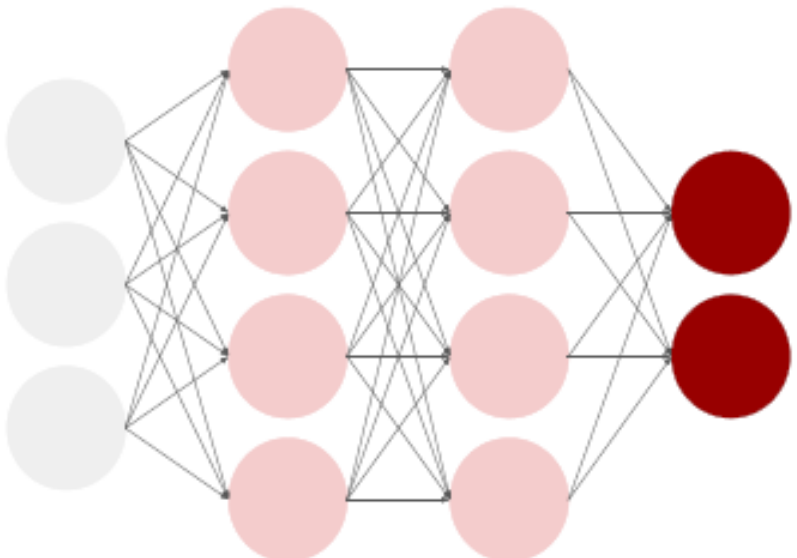
What is “**Language**” in Large Language Models?

- Human-understandable languages like English
- The models learn statistical patterns of the data
  - Example: Probability of "the" preceding the noun "cat" are very high
- These models can generate new text, mimicking Humans
- Applications:-



What is a “**Model**” in Large Language Models?

- A **model** refers to a mathematical representation of a real-world process
- It is essentially a program or an algorithm
  - Trained on a dataset to recognize patterns
  - Uses these patterns to make predictions or decisions without being explicitly programmed to perform the task
- ChatGPT uses GPT3 or GPT4 models



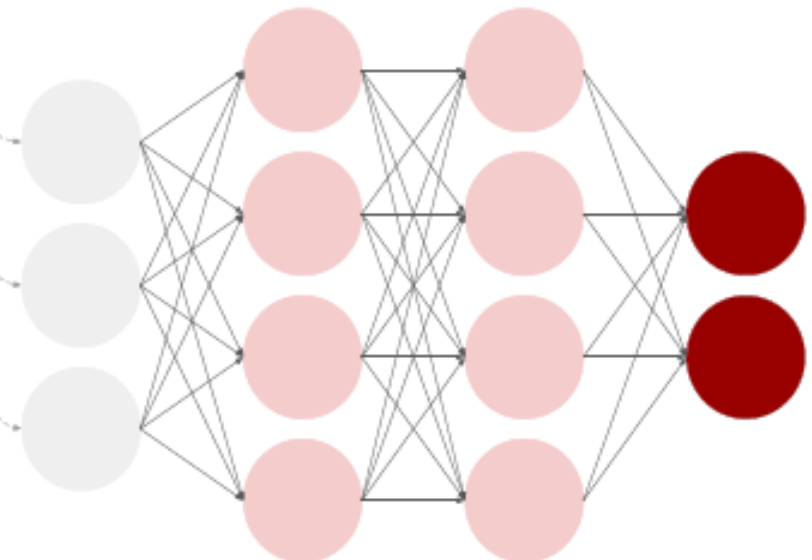
How a Large Language Model Works?

PRE-PROCESSING *(Example Text: She Sells Seashells)*

1) Normalize	2) Tokenize	3) Stop Words	4) Stemming & Lemmatization
she sells seashells	*["she", "sells", "sea","sh","ells"]*	*["sells", "seashells"]*	*["sell", "seashell"]*

- Text Normalization is the process of converting text to a standard format, such as to lowercase, removing special characters, and converting numbers to their written form
- Tokenization is the process of breaking down text into individual units, such as words or phrases. This is an important step in preparing text data for NLP tasks
- Stop Words are common words that are usually removed during text processing, as they do not carry much meaning and can introduce noise or affect the results of NLP tasks. Examples of stop words include "the," "a," "an," "in," and "is"
- Stemming and Lemmatization are techniques used to reduce words to their base form. This helps reduce the dimensionality of the data and improve the performance of models

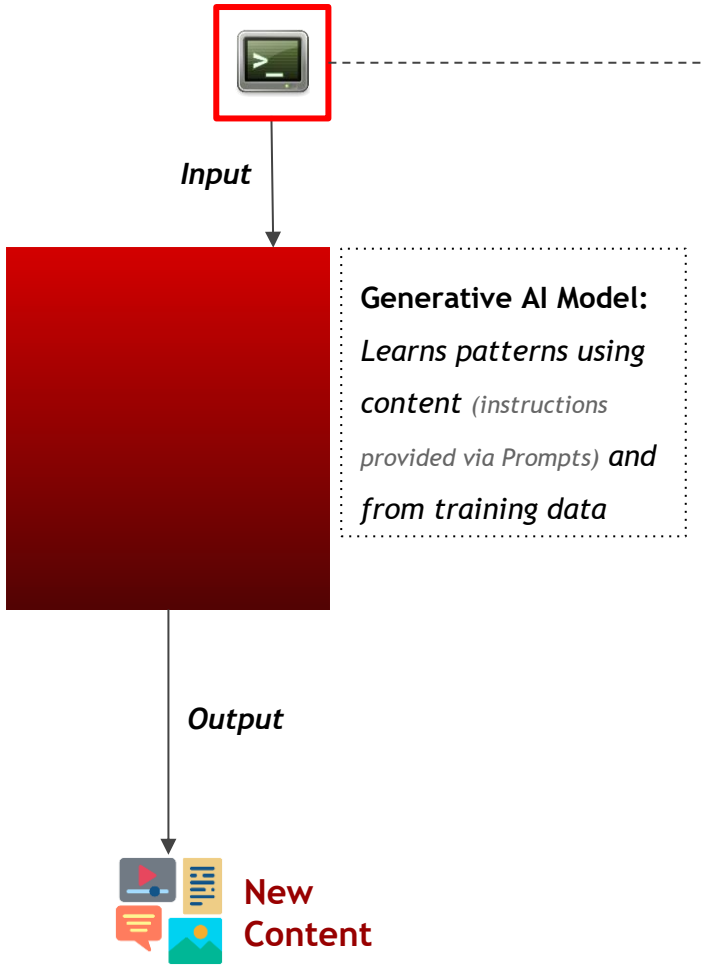
- LLMs are trained using a process called *unsupervised learning*
- This involves *feeding the model massive amounts of text data*, such as books, articles, and websites, and having the model learn the patterns and relationships between words and phrases in the text
- The model is then *fine-tuned on a specific task*, such as language translation or text summarization



Challenges with Large Language Model

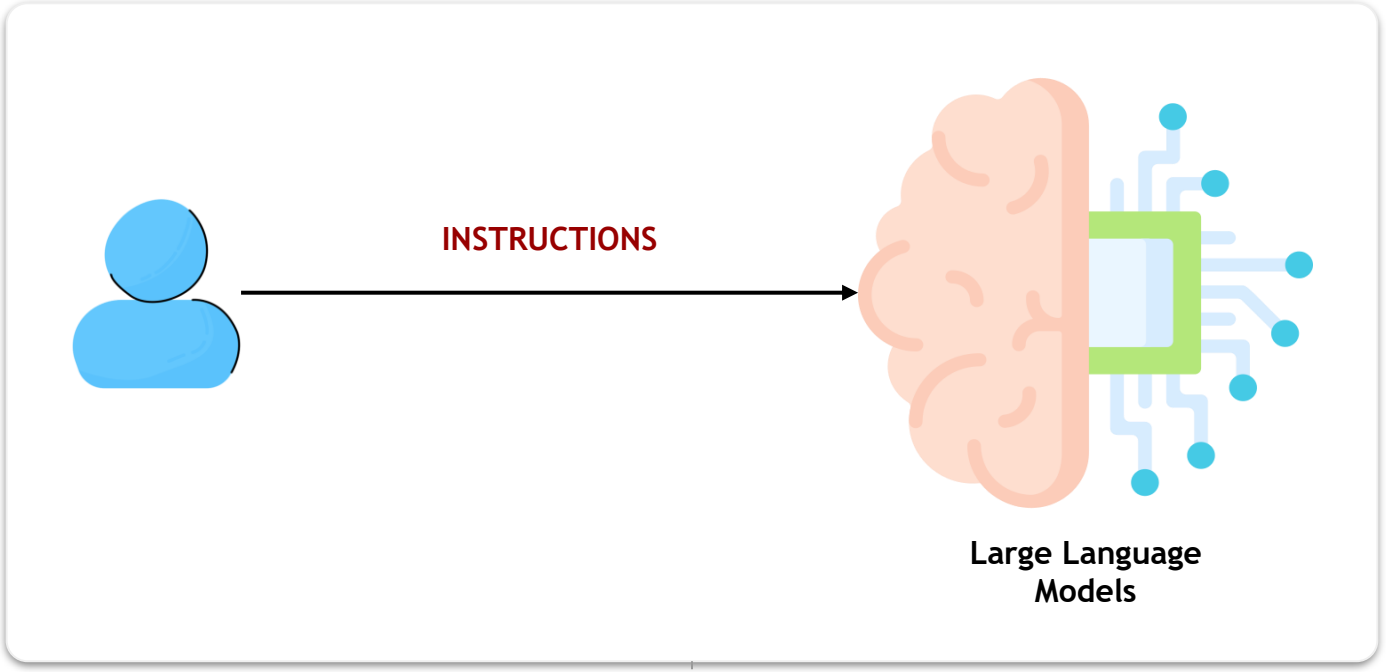
- One of the main challenges with LLMs is the **potential for offensive language**, as the models learn from the patterns found in the training data
- Unethical considerations, such as **gender and racial biases**
- **Amount of computational resources** needed to train and run LLMs, which can be expensive and energy-intensive
- **Making up things** which are not even facts
- While large language models have shown impressive performance on a variety of NLP tasks, they may not perform as well on specific tasks, such as those that require a **deeper understanding of the underlying context**

What is a Prompt?



- *A prompt refers to a specific instruction, query, or input given to an AI model to generate a response or perform a task*
- *It serves as cue, guidance, or a starting point for the AI system to produce an output based on its learned knowledge and language comprehension*
- *The form and tone of a prompt may vary depending on the AI application and the desired outcome*
- *Well-defined, context-driven, and unambiguous prompts bring AI output closer to the intended results*

What is Prompt Engineering?



Prompt Engineering is an art of asking the right question to get the best response from an LLM

What is Prompt Engineering?

Developer

As per me,

PROMPT

Code blocks, individual lines of code, or natural language comments a developer writes to generate a specific suggestion from GitHub Copilot

PROMPT ENGINEERING

Providing instructions or comments in the IDE to generate specific coding suggestions

CONTEXT

Details that are provided by a developer to specify the desired output from a generative AI coding tool

What is Prompt Engineering?

ML Researcher

As per me,

PROMPT

Compilation of IDE code & relevant context (IDE comments, code in open files, etc.) that is generated by algorithms & sent to the model of a Gen AI coding tool

PROMPT ENGINEERING

Creating algorithms that will generate prompts (compilations of IDE code and context) for a large language model

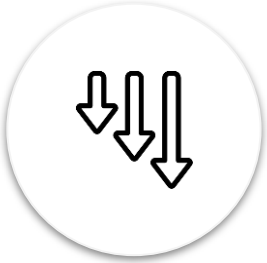
CONTEXT

Details (like data from your open files & code you've written before) that algorithms send to a large language model (LLM) as additional information about the code

Why is Prompt Engineering Important?

Improve the Accuracy

Improve the usefulness of AI generated content



Reduce the risk of generating harmful and biased content

Key elements of a Prompt

Let’s look at aspects that make up a good prompt:

Task

Clearly specify the desired action or response from the model

Context

Provide relevant background information to help the model understand the setting or scenario

Exemplar

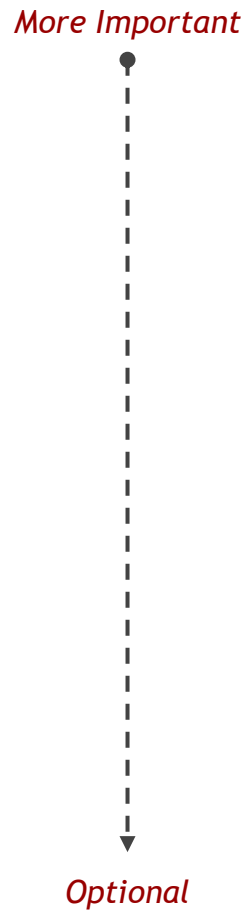
Offer a sample of the desired output to guide the model

Persona

Define the character or role the model should respond to

Format

Specify the structure or style of the response



Key elements of a Prompt

Let’s look at aspects that make up a good prompt:

Task

Always start the Task sentence with an Action Verb, e.g.

- ***GENERATE***
- ***GIVE***
- ***WRITE***
- ***ANALYZE***
- ***and many more …***

Key elements of a Prompt

Let's look at aspects that make up a good prompt:

Context

- *What is the user's background?*
- *What does success look like?*
- *What environment are they in?*

Create a Python script for an e-commerce website to automate customer order processing. Keep in mind that the user is a mid-level software engineer familiar with web development and APIs but new to handling large-scale transaction data. Success means the script is efficient, easy to maintain, and includes error handling for failed API calls. The user is working in a fast-paced startup environment with limited resources and a tight deadline.

Key elements of a Prompt

Let’s look at aspects that make up a good prompt:

Exemplar

Exemplars are not necessary for every prompt but including them as relevant examples greatly improves the quality of your output

Develop a REST API using Flask that handles user authentication for a web application. The user is a backend developer new to Flask but familiar with Django. Success involves creating an API with endpoints for registration, login, and logout. Provide an example of an API response for a successful login, including the structure of the JSON data and the status codes. Additionally, include sample error responses to demonstrate proper error handling.
Example Pseudocode:
First, import Flask and the necessary modules. Initialize a Flask application. Create a simple data dictionary to store user credentials:
from flask import Flask, request, jsonify
app = Flask(**name**)
users = {"test_user": "password123"}
Define a '/login' endpoint using the POST method. Retrieve the username and password from the incoming JSON request. Check if the username exists and matches the password. If successful, return a JSON response with a success status, a message, and an example token with a 200 status code.
If the credentials do not match, return an error response with a status of "error" and an "Invalid credentials" message, using a 401 status code.
Finally, run the Flask application in debug mode.
Expected Successful Response:
{ "status": "success",
"message": "Login successful!",
"token": "example_token_123"
}
Expected Error Response:
{ "status": "error",
"message": "Invalid credentials."
}

Key elements of a Prompt

Let’s look at aspects that make up a good prompt:

Persona

Who do you want the AI to be? Think of someone you wish you had instant access to with the task you’re facing

IF YOU’RE A SENIOR BACKEND ENGINEER

Imagine you are a senior backend engineer with over 10 years of experience in building secure and scalable web applications.

You've worked extensively with Flask, Django, and REST API design. I'm a mid-level developer building a REST API for user authentication in Flask. Guide me as a mentor would, helping me design the login endpoint to handle user credentials securely, implement proper error handling, and return appropriate JSON responses. Break down complex concepts into simple steps, and provide best practices to ensure the API is robust and secure.

Key elements of a Prompt

Let's look at aspects that make up a good prompt:

Format

Set the structure of the output from GPT-Models

Create a detailed guide for building a user authentication REST API using Flask. Structure the response into the following sections:

1. ***Introduction**: Briefly explain the purpose of user authentication and why Flask is a good choice.*
2. ***Requirements**: List all the libraries and tools needed to build the API.*
3. ***Step-by-Step Instructions**:*
    - *Include a clear breakdown of each step with explanations.*
    - *Provide code snippets for each part, starting from setting up the Flask app, creating endpoints for registration, login, and logout.*
4. ***Code Walkthrough**: Explain the key parts of the code, focusing on how authentication is handled.*
5. ***Best Practices**: Highlight important considerations such as security measures, error handling, and scalability.*
6. ***Sample API Responses**: Show example JSON responses for successful login, registration, and error scenarios.*
7. ***Conclusion**: Summarize the key takeaways and next steps.*

Ensure each section is clearly labeled and presented in a concise manner. Use simple language and provide comments within code snippets for better understanding

Key elements of a Prompt

Let's look at aspects that make up a good prompt:

You are a senior software architect with over 15 years of experience in developing secure and scalable web applications. I am a mid-level developer, working in a fast-paced startup environment, and I need to create a REST API for user authentication using Flask. I have experience with Django but am new to Flask and handling large-scale authentication systems.

Your task is to guide me through building a simple but secure authentication API. Please provide clear, step-by-step instructions. The goal is to ensure the API has endpoints for registration, login, and logout, implements token-based authentication, and includes proper error handling.

Structure the response as follows:

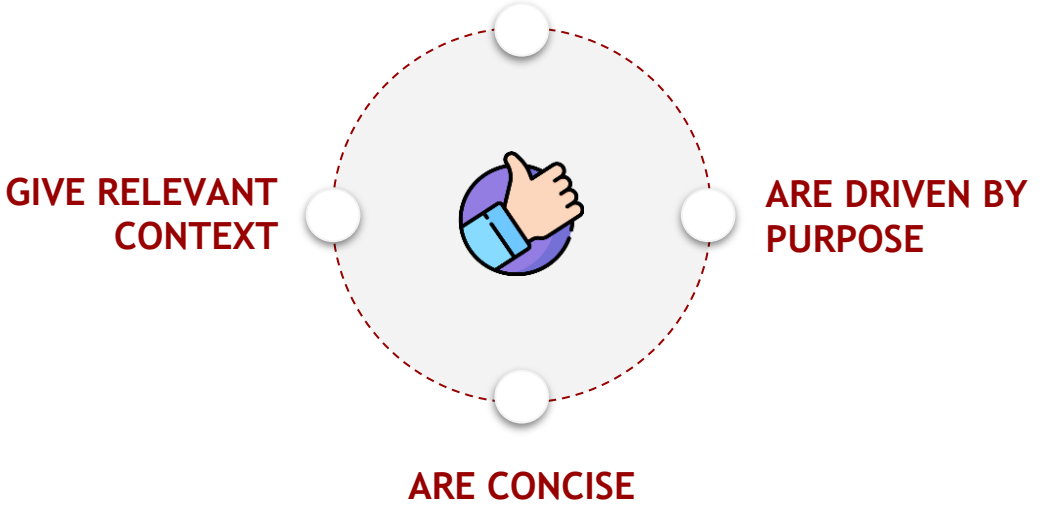
- ***Introduction**: A brief overview of user authentication and why Flask is suitable for this task.*
- ***Requirements**: List of libraries and tools needed.*
- ***Step-by-Step Implementation**: Break down the process with code snippets and explanations. Include comments within the code for clarity.*
- ***Best Practices**: Advice on security measures, error handling, and scalability.*
- ***Sample Code Response**: Provide an example response for a successful login and an error case.*

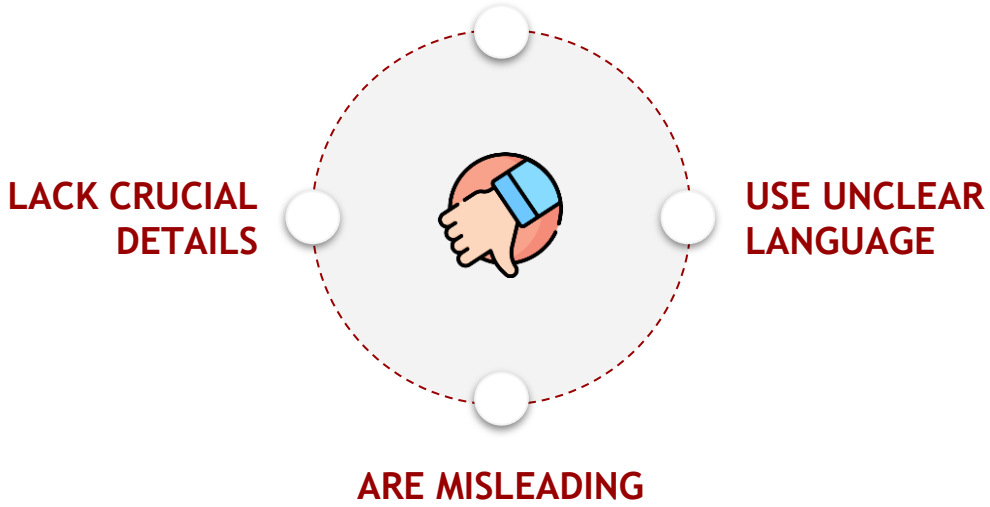
For the 'Step-by-Step Implementation' section, include a sample code snippet. For example, when implementing the login endpoint, show how to handle user input, validate credentials, and return a JSON response with a success message and token

PERSONA CONTEXT TASK FORMAT EXEMPLAR

Good Prompts vs. Bad Prompts

The quality of the prompt significantly influences the output generated by the model. Here are the traits of Good & Bad Prompts:





Tips for designing Prompts

Write Clear Instructions

Help me build an authentication system

Guide me through building a user authentication system in Flask. Include steps for creating endpoints for user registration, login, and logout. Use token-based authentication and include code examples for each step. Make sure to explain how to handle errors and security best practices

Change your prompt as per following to make it clearer:

- *Include details in your query to get more relevant answers*
- *Ask the model to adopt a persona*
- *Use delimiters to clearly indicate distinct parts of the input*
- *Specify the steps required to complete a task*
- *Provide examples*

Tips for designing Prompts

Ask to provide reference text when required

Explain data encryption

Explain data encryption as it relates to API security. Provide reference examples from standard security guidelines like OWASP if possible. Include practical examples for implementing encryption in a Python application

Change your prompt like this to make it clearer:

- *Instruct the model to answer using a reference text/website*
- *Instruct the model to answer with citations from a reference text*

Techniques in Prompt Engineering

Here, we see more intricate strategies that require a deeper understanding of the model's behavior

ZERO-SHOT PROMPTING

This technique involves providing the model with a task it hasn't seen during its training. It tests the model's ability to generalize and produce relevant outputs without relying on prior examples

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

Neutral

Techniques in Prompt Engineering

Here, we see more intricate strategies that require a deeper understanding of the model's behavior

FEW-SHOT PROMPTING

Here, the model is given a few examples (shots) to guide its response. By providing exemplars, the model can better understand and generate the desired output. For example, showing a model several examples of translated sentences before asking it to translate a new one

Craft a compelling tagline for social media introducing our upcoming product, a sustainable packaging initiative for our soft-drink line. Highlight the eco-friendly features and emphasize the positive impact on the environment.
Example1: Every sip is a step towards a greener future
Example2: Good for you, great for the Earth
Example3: Because flavor shouldn't cost the Earth. Share the taste, share the planet

Sip sustainably, savor responsibly

Techniques in Prompt Engineering

Here, we see more intricate strategies that require a deeper understanding of the model's behavior

CHAIN OF THOUGHTS (COT)

It involves guiding the model through a series of reasoning steps. By breaking down a complex task into intermediate steps or a "chains of reasoning," the model can achieve better language understanding and more accurate outcomes

Process the problem step-by-step and answer the question asked:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9


THANK YOU