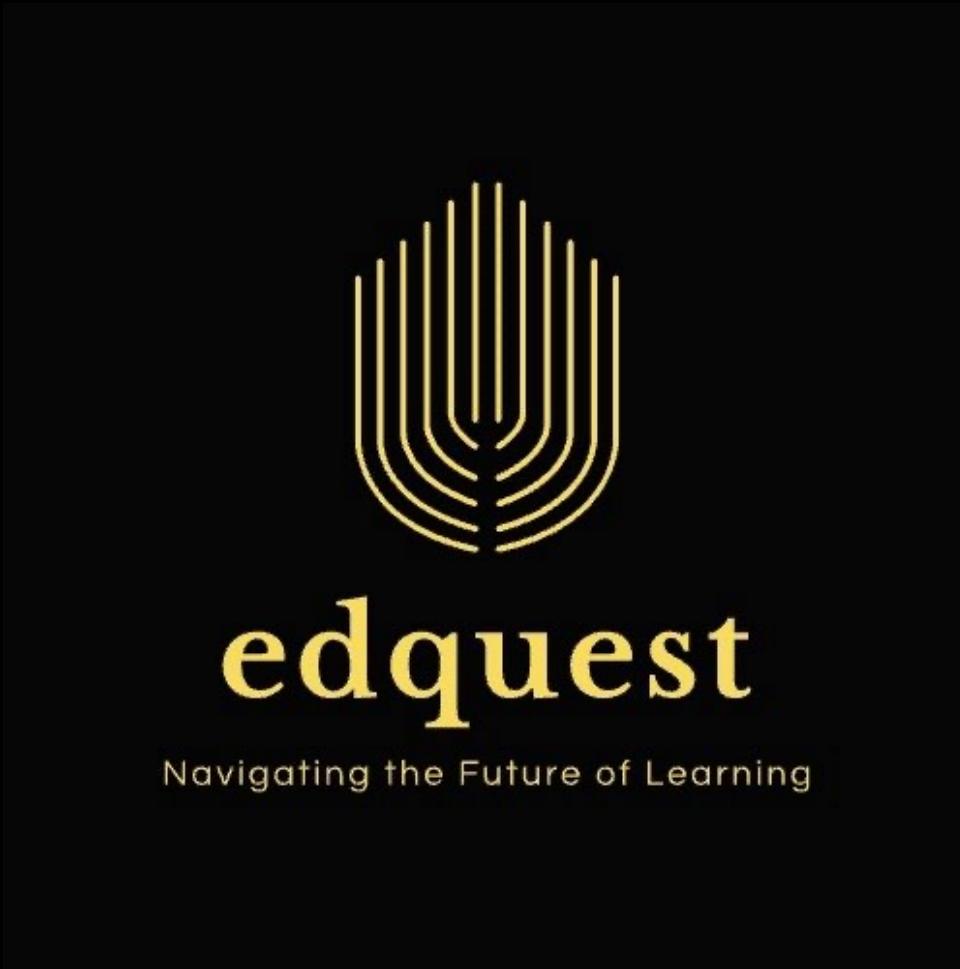


The Generative AI Revolution: A Month-Long AI Exploration





Agenda

- › Impact of GenAI across different Industry
- › Task performed by GenAI and Impact
- › Concern about AI
- › Responsible AI
- › Opportunity created by AI



“AI is impacting various industries by automating the entire job or part of job.”



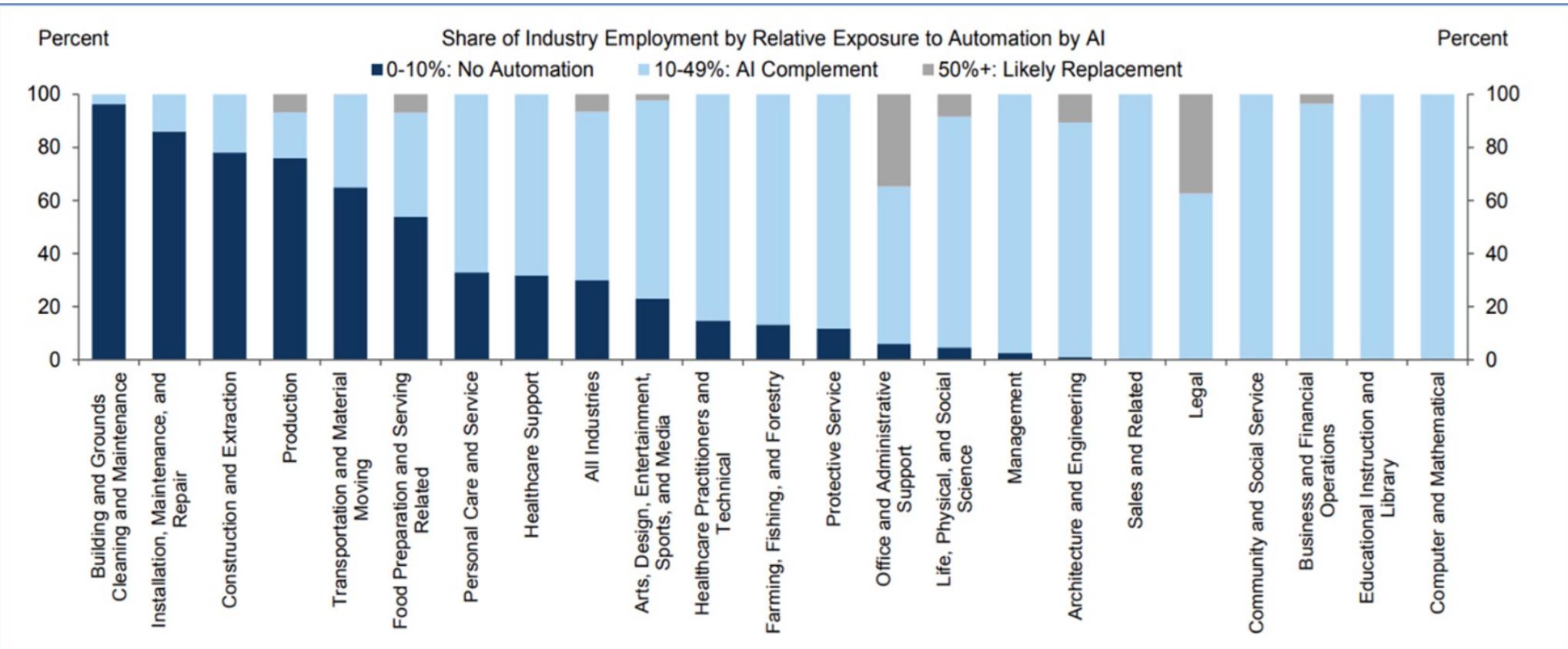
Impact of GenAI across Industry

01	Technology [\$240 - 460B]	<ul style="list-style-type: none">Software development and testingCybersecurityWeb Development and design etc.
02	Retail [\$240 - 390B]	<ul style="list-style-type: none">Customer experience and serviceMarketing and salesSupply chain and inventory management etc.
03	Banking [\$200 - 340B]	<ul style="list-style-type: none">Customer service [Chabot and virtual assistant]Risk management and complianceCredit score and lending
04	Healthcare	<ul style="list-style-type: none">Diagnostic and imagingDrug discovery and developmentPatient care and monitoring
05	Education	<ul style="list-style-type: none">Personalized learningCurriculum developmentEducation administration

Source: McKinsey Report



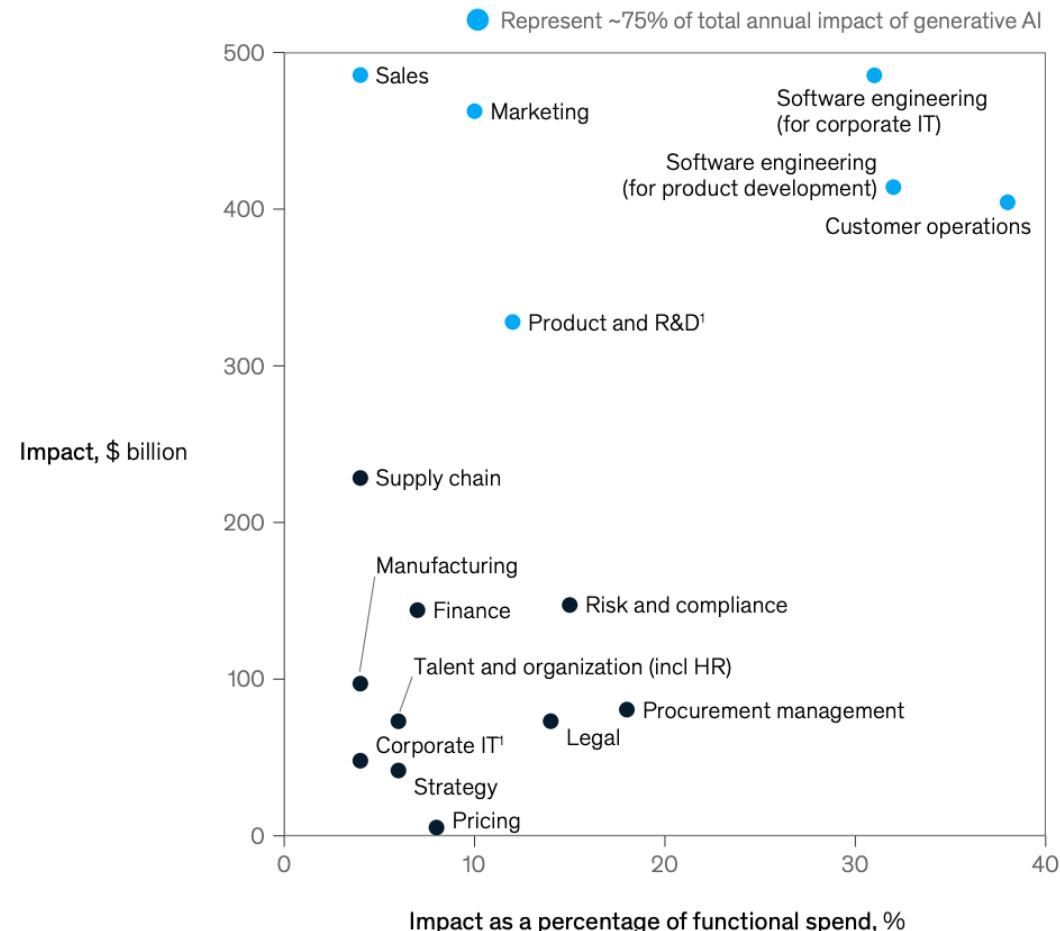
Impact of GenAI across Function



Source: Goldman Sachs Report



Impact of GenAI across Function



Source: McKinsey Report

π



Navigating the Future of Learning



Task Performed by Generative AI

- › Generation
- › Translation
- › Summarization
- › Search or Extraction
- › Classification or Tagging
- › Editing or Re-writing
- › Analysis or Interpretation

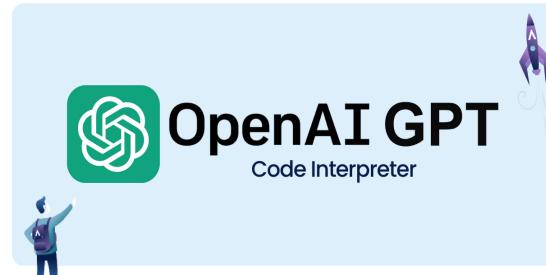


Task: Generation

- › Text Generation
- › Image Generation
- › Video Generation
- › Code Generation



Code Generation





Code Generation: Example

**You**

Write Python function code to swap two variable without using any temp variable. No need to put comment just write code.

**ChatGPT**

Here's the Python function to swap two variables without using a temporary variable:

python

Copy code

```
def swap_variables(a, b):
    return b, a

# Example usage
x = 5
y = 10
x, y = swap_variables(x, y)
```



Text Generation

- › Product description
- › Website write-up
- › Brand communication
- › Generate policy
- › Product document



Image Generation

- › Generate image for product portfolio
- › Marketing creative
- › Advertisement
- › Product design
- › E-commerce



Image Generation: LLM/Tool

- › DALLE-3
- › Midjourney
- › Adobe Firefly
- › Google Bard



Image Generation: Example



You

Give me a beautiful image of TajMahal during morning.

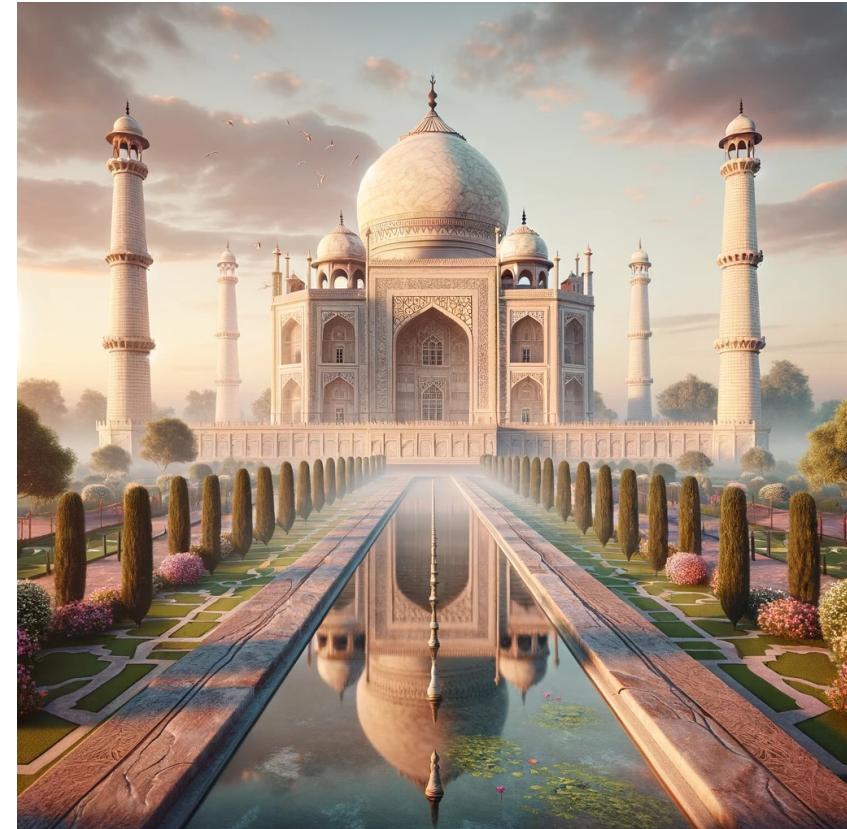




Image Generation: Example



You

Give me a beautiful image of TajMahal during night.





Video Generation

- › Generate video for advertisement
- › Product education
- › Product training



Link: https://youtu.be/5WECsbqAQSk?si=Hy6d4hfz2Oq_1LzA

π



Navigating the Future of Learning



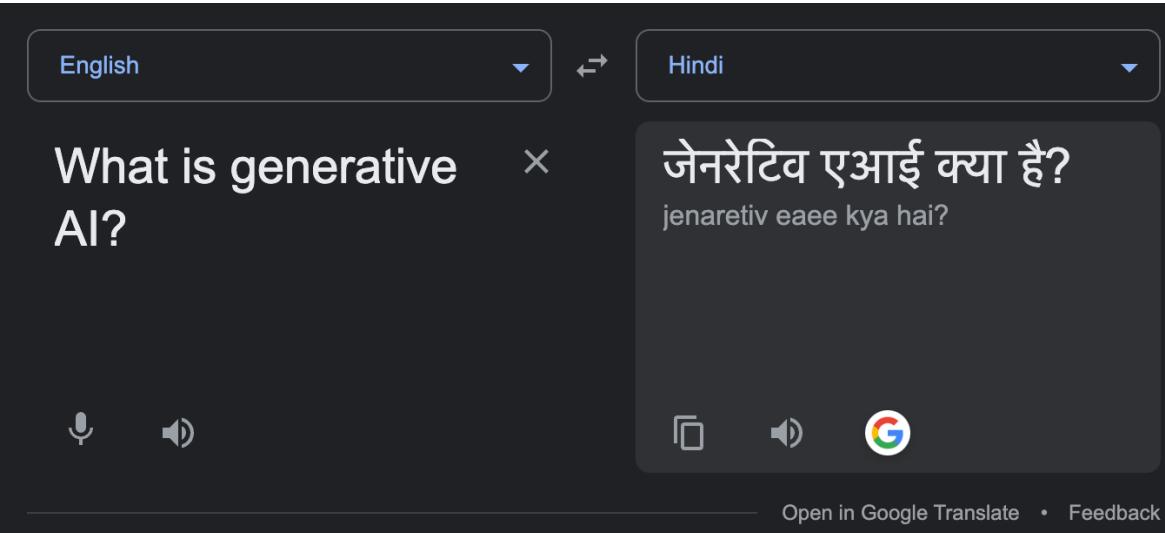
Task: Translation

Translate a piece of content from one language to another language. This could be text or audio.





Translation: Example



Indian language AI tool ‘Bhashini’ used to translate Prime Minister Narendra Modi’s speech

Prime Minister Narendra Modi used the AI-powered Bhashini tool so that his Tamil audience could hear a real-time translation of his Hindi speech

December 18, 2023 11:31 am | Updated 12:25 pm IST – VARANASI

THE HINDU BUREAU



Task: Summarization

Summarize large piece of text or document into smaller piece by retaining the key point:

- › Summarize email, documents and meeting notes
- › Summarize complex legal, policy and regulation documents
- › Summarize feedbacks, description and report



Task: Search or Extraction

Search for or extract relevant text, numbers or data from a large corpus of text:

- › Extract customer journey and life-cycle for product
- › Extract relevant information from knowledge base
- › Extract relevant piece of information from large document



Task: Classification or Tagging

Classify, tag or cluster text or documents as per your defined categories:

- › Classify review
- › Classify news and blog
- › Classify product



Task: Editing or Rewriting

Edit or re-write existing piece of text in such a way that suit your requirements:

- › Edit text to correct the grammar
- › Re-write to change the style and tone
- › Edit or re-write campaign for personalize touch in marketing



Task: Analysis or Interpretation

Analyse or interpret a large piece of text to get the relevant insight:

- › Analyse customer call and feedback to find customer issue
- › Analyse and interpret financial data
- › Analyse code for root cause and automation

π



Navigating the Future of Learning



Concern about AI

- › Amplifying humanity's worst impulses
- › Job loss like radiologist
- › Human extinction



Dimensions of Responsible AI

- › **Fairness:** Ensure AI does not amplify biases
- › **Transparency:** AI decisions should be understandable
- › **Privacy:** Protect user data and ensure confidentiality
- › **Security:** Safeguard AI systems from malicious attacks
- › **Ethical Use:** Ensuring AI is used for beneficial purpose



Opportunity Create by AI

- › Prompt Engineering
- › Generative AI Developer
- › AI-assisted Designer
- › Digital Artist
- › Content Creator using AI



Thank You!



Mokarbeen Ansari
www.linkedin.com/in/mokarbeenansari/

The Generative AI Revolution: A Month-Long AI Exploration



edquest

Navigating the Future of Learning

Session-2

Week 1 Day 2: 4th Feb 2024



Prompt Engineering

What is prompting
Importance of prompting
Structure of a good prompt
Enhancing LLM capabilities using advanced prompting techniques

π





What is a prompt?

A prompt refers to the **input provided to the model to generate a response**. The prompt can be a question, instruction, or any form of input text that guides the model in *producing relevant output*.

What is prompt engineering?

Prompt engineering is the process of **designing and optimizing prompts** used in natural language processing (NLP) models, such as ChatGPT, chatbots, or virtual assistants. This involves **crafting prompts that are clear, concise, and effective in eliciting the desired response**.



What is prompt priming?

"Priming" refers to the practice of **providing some initial input** to the model before generating a response. The initial input is intended to guide the model toward generating a response that is more relevant and coherent with the user's intended output.

Example:

Without priming: Where should I go on my next vacation?

With priming: I'd like to go on a trip with my wife and kids. The location should be tropical, we'd love to be close to the beach, I'd like the flight to be one direct flight to and from LAX, and I have a travel budget of \$5,000. **Where should I go on my next vacation?**

Priming Output: One suggestion for your tropical family vacation within a **\$5,000 budget** could be **Cancun**, Mexico. **Direct flights** from LAX are available, and you can find affordable all-inclusive packages with prices starting at around \$1,000 per person for a week-long stay. Popular **family activities** in Cancun include snorkelling, visiting ancient ruins, and exploring **local beaches** and restaurants.

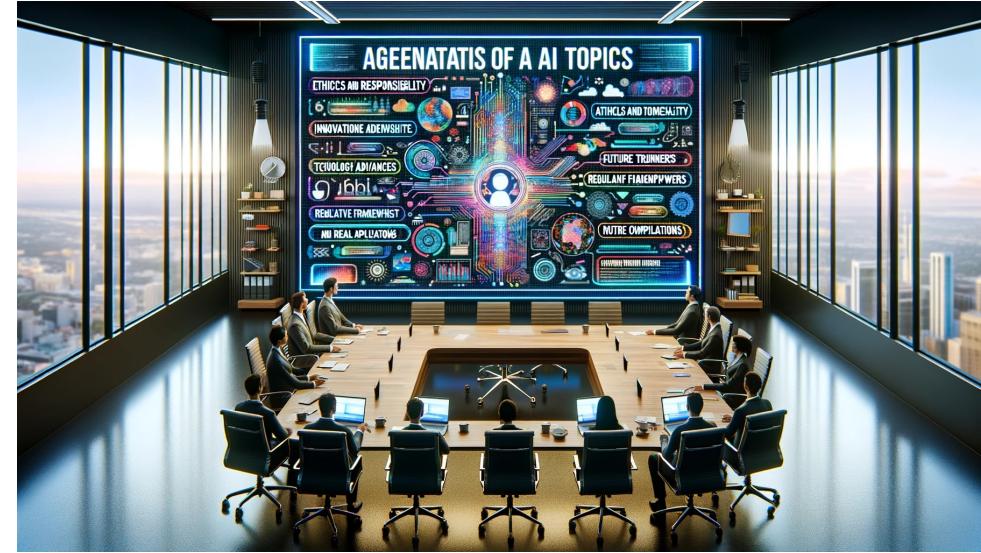


Engineering a Good Prompt

Prompting Principle – 1 *(Providing Clear instructions)*

Five components of a good prompt:

1. Explicitly state the **task** to be performed to get the required output
2. **Role** of the language model for performing the task
3. **Context** in which the task must be done
4. **Guidelines** on how to perform the task
5. The desired **output format** of the ideal response





Applying Prompt Engineering for Language Tasks

Task

The task is to write compelling ad copy for a new laptop model highlighting its features, benefits, and special offers or pricing on our e-commerce platform.

As a language model, I will serve as a copywriter who understands marketing strategies and consumer psychology. I will craft the ad copy to be engaging, informative, and persuasive, using language that resonates with a diverse audience of potential buyers.

Role

The ad copy is intended for an online e-commerce website targeting consumers interested in high-configuration laptops. The Laptop features a Powerful Processor, Dedicated Graphics Card, High-Resolution Display, Large and Vibrant Screen, Long Battery Life, Fingerprint Sensors or Facial Recognition, Fast Storage and many more. The context also includes a competitive market with tech-savvy customers who value innovation and quality.

Context

The ad copy should:

####

Begin with an attention-grabbing headline.

Include key selling points like exceptional Performance, Premium Build and Design, and Future-Proofing.

Mention any discounts, warranties, or limited-time offers.

Have a call to action that encourages immediate purchase.

Be concise, with a friendly and inviting tone.

Avoid technical jargon that might confuse the average customer.

####

Guidelines

The ad copy should be formatted as follows:

####

A catchy headline

A brief introduction

A bullet list of key features and benefits

A closing sentence with a call to action

####

Output Format

The ad copy should be no more than 200 words to ensure it is quick to read and easy to understand



Output

Unleash the Future: Power and Elegance Combined!

🌟 Introducing the Ultimate High-Performance Laptop 🌟

In a world where technology is constantly evolving, step ahead with our new high-configuration laptop. Designed for those who demand excellence, it's more than just a laptop – it's a revolution in your hands.

Key Features and Benefits:

- 🚀 **Exceptional Performance:** Experience lightning-fast speeds with our cutting-edge processor and dedicated graphics card. Say goodbye to lag and hello to seamless multitasking.
- 💎 **Premium Build & Design:** Sleek, sturdy, and stylish. A design that speaks volumes about your taste for quality and elegance. Take it anywhere, turning heads as you go.
- 📺 **Vibrant High-Resolution Display:** Dive into a large, vibrant screen that brings visuals to life. Immerse yourself in stunning clarity and vivid colors for work and play.
- ⏳ **Long Battery Life:** Stay unplugged longer. Our extended battery life means more work, more play, and less worry about finding the nearest outlet.
- 🔒 **Advanced Security:** Keep your data safe with cutting-edge fingerprint sensors and facial recognition technology – security at your fingertips.
- 💻 **Fast Storage:** Store more and access it faster. Enjoy ample space with rapid response times, keeping everything you need at your fingertips.

Special Offers:

- 💰 **Exclusive Discount:** Grab a 10% discount for a limited time only!
- 🛡️ **Extended Warranty:** Peace of mind with an additional 1-year warranty at no extra cost.

Ready to Elevate Your Experience?

🛒 Don't miss out! **Order now** and step into the future of computing. Your journey to excellence starts here!



Applying Prompt Engineering for Code-Related Tasks

Develop a script or application using Slack's API that will sync the company's website with the Slack users. If a user is added to Slack, he/she should be added automatically to the company's website. If the user is removed from Slack, he or she should be removed from the Company's website.

Task

As a language model, my role is to act as a software developer who can **outline the steps necessary to write this code**, including explaining authentication with Slack's API, handling requests and responses etc.

Role

The code will be used to automatically add or remove the users to the company's website when the IT team adds or removes a user from Slack, that user should be automatically added or removed from the company's website respectively.

Context

The steps to perform this task would include:

####

Registering the application with Slack to obtain the necessary credentials (Client ID and Client Secret).

Using the credentials to authenticate

Setting up a webhook or using the Slack API's to update the website.

Guidelines

Writing a function to update the company's website

Handling errors and ensuring the message format is compatible with Slack's formatting options.

####

Output Format

The ideal response should be a code snippet or a set of code snippets that are ready to be implemented, with comments explaining each section of the code.



Applying Prompt Engineering for Code-Related Tasks (*ChatGPT output*)

Creating a full-fledged application to synchronize a company's website user list with Slack's user list is a complex task that involves several components. Below, I will outline a basic Python script using Flask to handle the Slack Events API and update the user list on the company's website accordingly.

Step-by-step guide and code snippets:

1. Registering Your Application with Slack:

1. Go to your Slack workspace's app configuration page to create a new app.
2. Add permissions to access user data and subscribe to user change events.
3. Obtain the Client ID, Client Secret, and Signing Secret.

2. Setting Up OAuth for Slack Authentication:

1. Implement an OAuth flow to get the necessary access token to use Slack APIs.

3. Setting Up the Flask Application:

1. Create a Flask application to receive events from Slack.
2. Verify the requests from Slack using the Signing Secret.

4. Handling User Addition and Removal:

1. Listen for user addition and removal events.
2. Update the company's website using its API.

5. Error Handling and Formatting:

1. Add error handling to manage unexpected issues.
2. Ensure the message format aligns with Slack's requirements.



Applying Prompt Engineering for Code-Related Tasks (*high-level code outline*)

python

```

from flask import Flask, request, jsonify, abort
import hmac
import hashlib
import requests
import os

# Initialize Flask app
app = Flask(__name__)

# Slack credentials obtained during app registration
SLACK_SIGNING_SECRET = os.environ['SLACK_SIGNING_SECRET']
SLACK_ACCESS_TOKEN = os.environ['SLACK_ACCESS_TOKEN']

# Your company's API endpoint to update the user list
COMPANY_UPDATE_USER_API = 'https://yourcompany.com/api/update-user'

def verify_slack_request(slack_signature, timestamp, body):
    """
    Verify the request is coming from Slack.
    """
    basestring = f"v0:{timestamp}:{body}".encode('utf-8')
    my_signature = 'v0=' + hmac.new(
        SLACK_SIGNING_SECRET.encode('utf-8'),
        basestring,
        hashlib.sha256
    ).hexdigest()
    return hmac.compare_digest(my_signature, slack_signature)

```

Copy code

```

def update_company_user_list(user_id, action):
    """
    Update the user on the company's website.
    """
    response = requests.post(
        COMPANY_UPDATE_USER_API,
        json={'user_id': user_id, 'action': action},
        headers={'Authorization': f'Bearer {COMPANY_API_A'}
    )
    return response.ok

```

```

@app.route('/slack/events', methods=['POST'])
def slack_events():
    """
    Endpoint for Slack's Events API.
    """

    # Verify Slack request
    slack_signature = request.headers.get('X-Slack-Signature')
    timestamp = request.headers.get('X-Slack-Request-Timestamp')
    if not verify_slack_request(slack_signature, timestamp, request.get
        return abort(403)

    event_data = request.json
    if 'challenge' in event_data:
        # Respond to the Slack challenge request
        return jsonify({'challenge': event_data['challenge']})

    # Handle user change events
    if event_data['event'][['type']] == 'user_change':
        user_id = event_data['event'][['user']]['id']
        # Assume that the user has been added or updated
        if update_company_user_list(user_id, 'add'):
            return jsonify({'status': 'success'})

    # Slack does not send user removed events, so you would need to per
    # sync the user list and remove users not present in Slack anymore.

    return jsonify({'status': 'error'}), 400

if __name__ == '__main__':
    app.run(debug=True)

```



Engineering a Good Prompt

Prompting Principle – 2 *(Enhancing LLM Reasoning Capabilities)*

1. **Chain-of-Thought prompting** (*asking to approach step by step*)

2. **Few-Shot prompting** (*provide relevant and exhaustive examples*)





Chain-of-Thought prompting

Chain-of-thought prompting is a technique used in AI language models like GPT-3 and GPT-4 **to improve performance on reasoning tasks**. This approach involves prompting the model to generate intermediate steps or reasoning processes while solving a problem rather than directly aiming for the final answer.



Key Characteristics:

- 1. Explicit Reasoning:** The prompt includes the problem and a step-by-step breakdown of how one might logically solve it.
- 2. Clarity and Detail:** The reasoning steps are clear and detailed and follow a logical sequence.
- 3. Emulation of Human Problem-Solving:** The model is guided to emulate how a human might approach the problem, considering various aspects and possibilities.



Chain-of-Thought prompting Examples

Mathematical Problem:

1. **Standard Prompt:** "What is 35% of 240?"
2. **Chain-of-Thought Prompt:** "To find 35% of 240, first calculate 10% of 240, which is 240 divided by 10. Then, multiply this by 3 to get 30%. After that, find 5% of 240 by halving the value of 10%. Finally, add these values together to get 35%."
3. **Model's Response:** "10% of 240 is 24. So, 30% is $24 \times 3 = 72$. 5% of 240, which is half of 10%, is 12. Therefore, 35% of 240 is $72 + 12 = 84$."

Prompt: If there are 5 apples and you take away 3, how many do you have?

Chain-of-Thought:

1. You start with 5 apples.
2. The phrase 'take away 3' means you are taking 3 apples.
3. So, you now have 3 apples in your possession.
4. It's not about how many apples are left in the original place but how many you have taken.
5. Therefore, you have 3 apples.



Few-Shot prompting with Examples

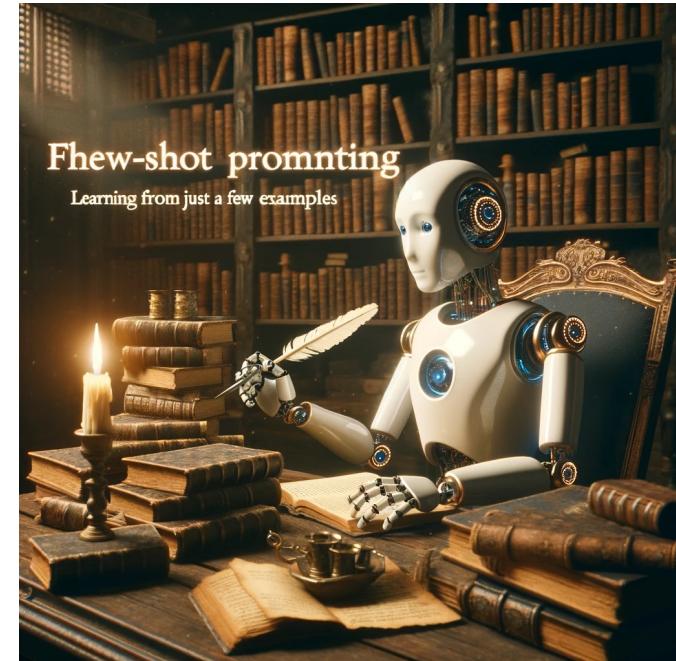
Few-shot prompting is an approach where an AI **model is given a small number of examples** (typically less than 10) to help it understand and perform a specific task. This method relies on the model's pre-trained knowledge and its ability to generalize from a few examples to new, unseen instances.

1. Sentiment Analysis:

- Example 1: "Sentence: 'I love sunny days.' Sentiment: Positive"
- Example 2: "Sentence: 'This movie is boring.' Sentiment: Negative"
- New Input: "Sentence: 'The cake tastes amazing.' Sentiment: ?"
Expected Output: "Positive"

2. Translation:

- Example 1: "English: 'Hello, how are you?' French: 'Bonjour, comment ça va?'"
- Example 2: "English: 'Thank you very much.' French: 'Merci beaucoup.'"'
- New Input: "English: 'Where is the library?' French: ?"
Expected Output: "Où est la bibliothèque?"





Few-Shot prompting Examples

3. Mathematical Problems:

- Example 1: "Problem: 'If I have 3 apples and eat one, how many are left?' Solution: '2 apples.'"
- Example 2: "Problem: 'A shirt costs \$20 and I have a \$5 coupon. How much do I pay?' Solution: '\$15.'"
- New Input: "Problem: 'If I have 5 oranges and my friend gives me 2 more, how many oranges do I have?' Solution: ?"
Expected Output: "7 oranges"

4. Grammar Correction:

- Example 1: "Incorrect: 'He don't know.' Correct: 'He doesn't know.'"
- Example 2: "Incorrect: 'She go to school every day.' Correct: 'She goes to school every day.'"
- New Input: "Incorrect: 'They has a cat.' Correct: ?"
Expected Output: "They have a cat."

*AI model uses the provided inputs and outputs to learn the pattern or the relationship between them. When a new input is given without an output, the model applies the **inferred pattern** to generate an appropriate output.*



Nuggets

Chrome extension: **YouTube Summary with ChatGPT**

<https://github.com/f/awesome-chatgpt-prompts>

Also Visit:

<https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>

<https://www.deeplearning.ai/short-courses/>



π

π



Thank You!



Vinit Narula

www.linkedin.com/in/vinitnarula/