

# INTRODUCTION TO PROMPT ENGINEERING

Yogesh Haribhau Kulkarni



# Outline

① GENAI OVERVIEW

② PROMPTENGG OVERVIEW

③ DEMO

④ CONCLUSIONS

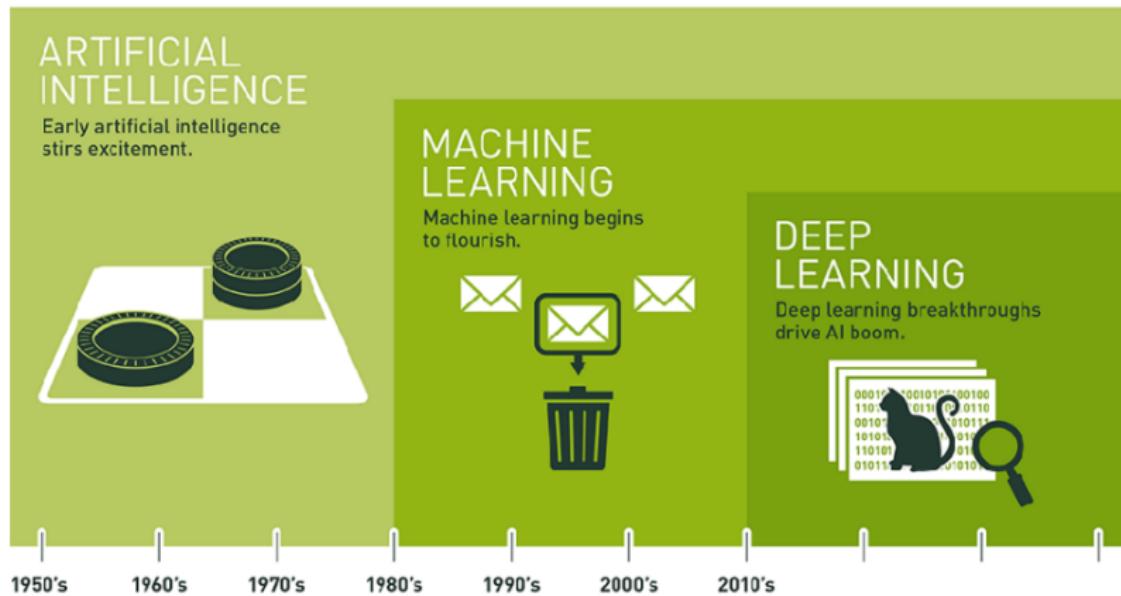
# Introduction to Generative AI

# Introduction

- ▶ What is Generative AI?
- ▶ What is not Generative AI?
- ▶ How is it related to AI-ML-DL?

# Relationship between AI, ML, DL

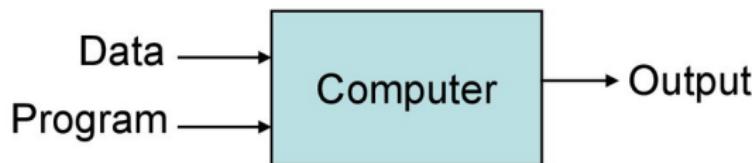
First, let's see what AI-ML-DL and relationship among them.



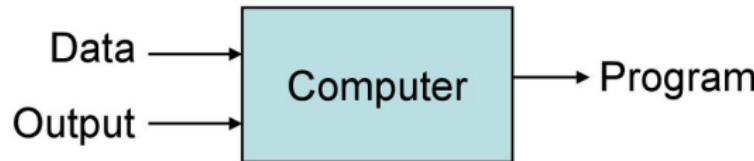
(Ref: <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>)

# Traditional vs. Machine Learning?

## Traditional Programming



## Machine Learning



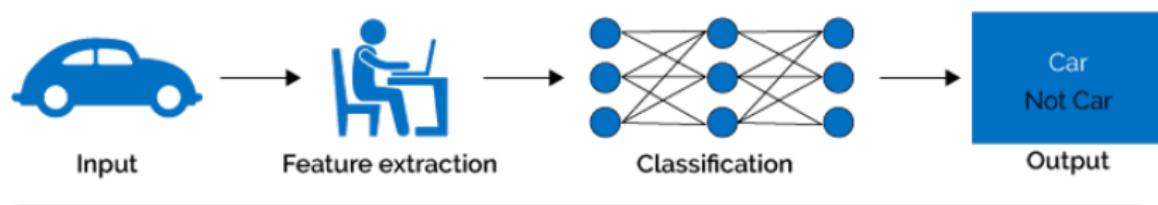
# Why Machine Learning?

- ▶ Problems with High Dimensionality
- ▶ Hard/Expensive to program manually
- ▶ Job \$\$\$

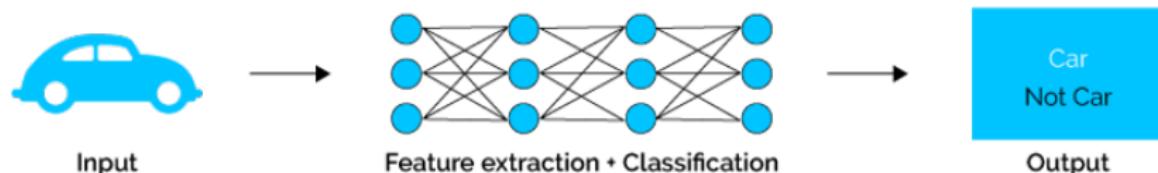
## ML vs DL: What's the difference?

Deep learning algorithms attempt to learn (multiple levels of) representation by using a hierarchy of multiple layers

### Machine Learning



### Deep Learning



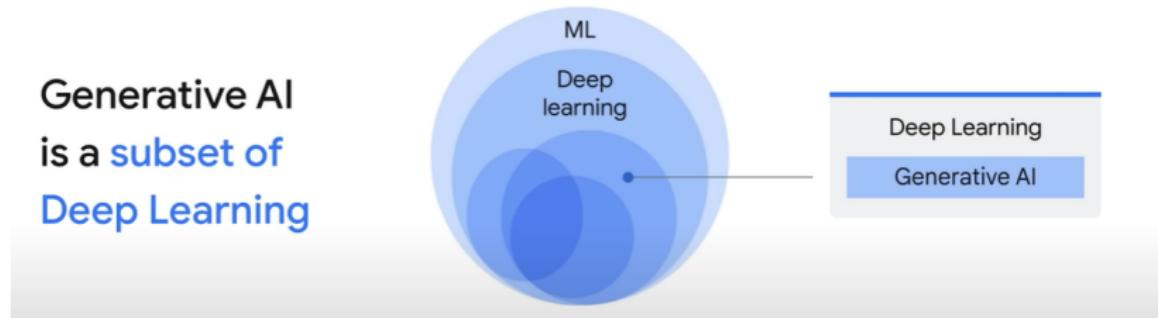
(Reference: <https://www.xenonstack.com/blog/static/public/uploads/media/machine-learning-vs-deep-learning.png>)

## Use Deep Learning When ...

- ▶ You have lots of data (about 10k+ examples)
- ▶ The problem is “complex” - speech, vision, natural language
- ▶ The data is unstructured
- ▶ Techniques to model ‘ANY’ function given ‘ENOUGH’ data.

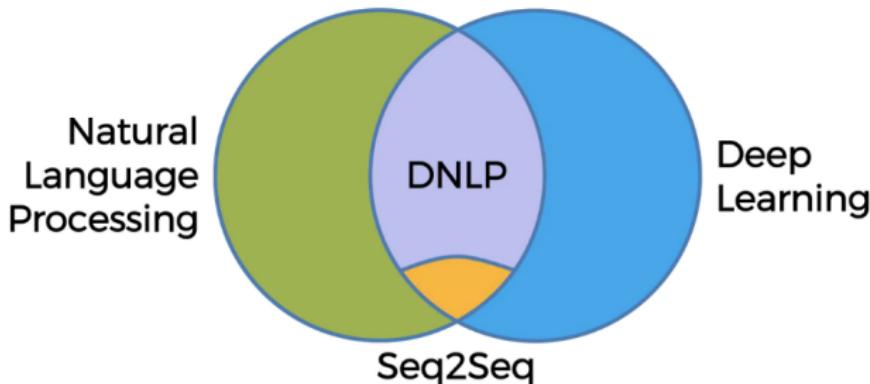
(Ref: Introduction to TensorFlow 2.0 - Brad Miro)

# What is Gen AI wrt AI, ML, DL



(Ref: Introduction to Generative AI - Google Cloud Tech)

## What is Deep NLP



(Ref: Deep Learning and NLP A-Z - Kirill Eremenko)  
(Note: Size is not indicative of importance)

Seq2Seq is heavily used technique of DNLP for sequence to sequence modeling, eg Translation, Q & A, etc. Thats the basis of Large Language Models (LLMs)

# Types of Approaches

## Deep Learning Model Types



### Discriminative

- Used to classify or predict
- Typically trained on a dataset of labeled data
- Learns the relationship between the features of the data points and the labels

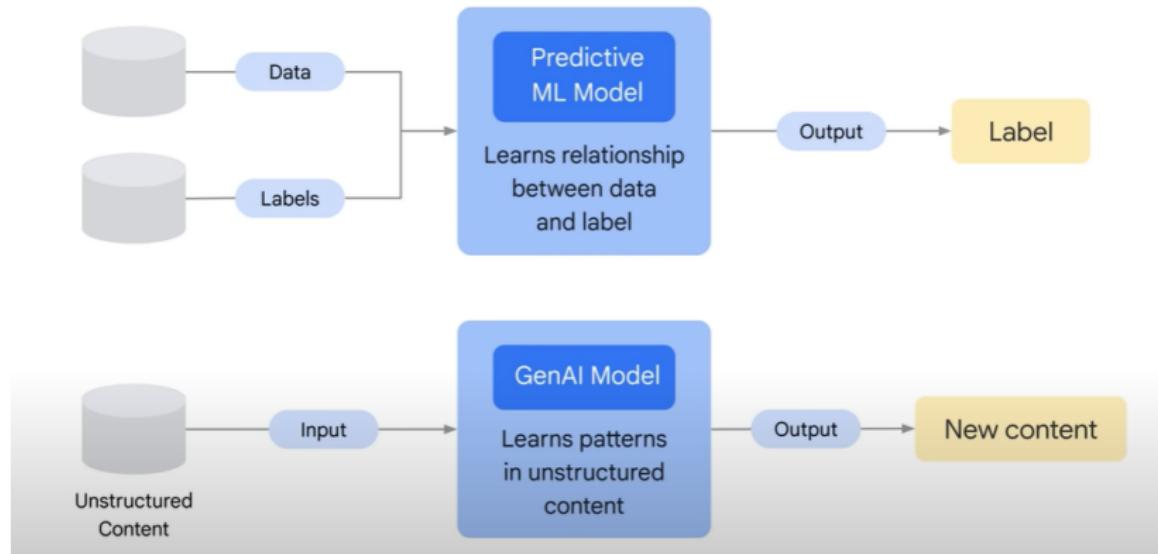


### Generative

- Generates new data that is similar to data it was trained on
- Understands distribution of data and how likely a given example is
- Predict next word in a sequence

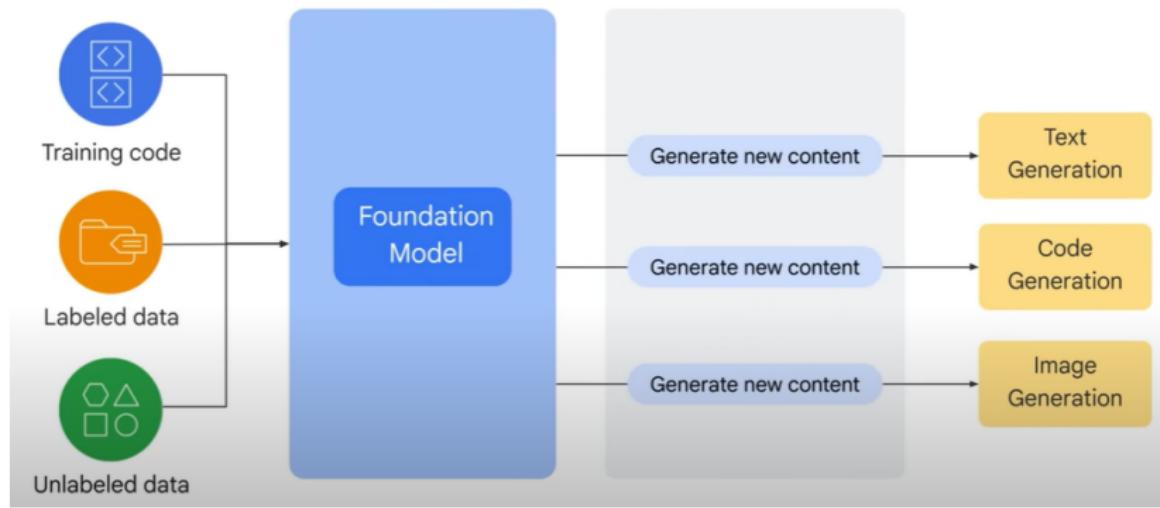
(Ref: Introduction to Generative AI - Google Cloud Tech)

## Types of Approaches



(Ref: Introduction to Generative AI - Google Cloud Tech)

# What is Foundation Model?



(Ref: Introduction to Generative AI - Google Cloud Tech)

## Same Problem, using different Technologies

## Difference across technologies, old to new

Lets see how the solutions to the problem of detecting a cat from images using traditional programming, deep learning, and generative AI, respectively.



## Traditional Programming

- ▶ Traditional programming involves writing explicit rules to detect a cat in images.
- ▶ Features like color, texture, and shape can be used to define these rules.
- ▶ However, designing accurate rules for complex patterns like cat detection can be challenging.
- ▶ It requires extensive domain knowledge and might not generalize well to different images.

```
cat:  
  type: animal  
  legs: 4  
  ears: 2  
  fur: yes  
  likes: yarn, catnip
```

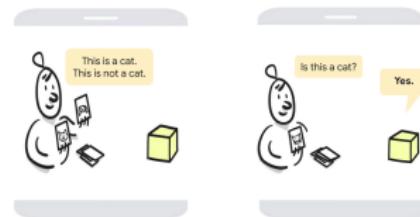


(Ref: Primer on LLM and Gen AI - Google Cloud)

YHK

# Deep Learning

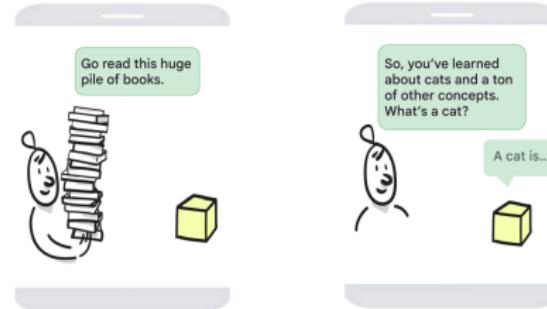
- ▶ Deep learning utilizes neural networks to automatically learn features for cat detection.
- ▶ Convolutional Neural Networks (CNNs) are particularly effective for image classification tasks.
- ▶ Large labeled datasets of cat images are used to train the network.
- ▶ The network learns to identify unique cat features and generalize them to detect cats in new images.
- ▶ Deep learning offers better accuracy and can handle complex patterns without explicit rule definition.



(Ref: Primer on LLM and Gen AI - Google Cloud)

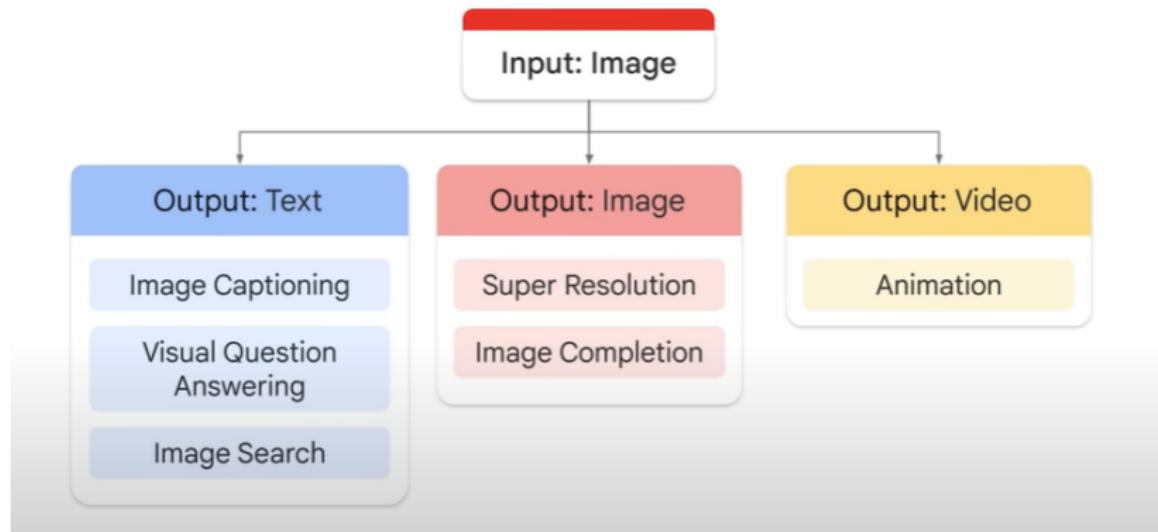
# Generative AI

- ▶ Generative AI focuses on generating new data, including images of cats.
- ▶ Generative Adversarial Networks (GANs) are used to generate realistic cat images.
- ▶ The GAN consists of a generator and a discriminator that compete against each other.
- ▶ The generator learns to generate increasingly realistic cat images, while the discriminator learns to distinguish real from generated images.
- ▶ The generated cat images can be used to augment datasets for cat detection models.



(Ref: Primer on LLM and Gen AI - Google Cloud)

# Modalities in Generative AI

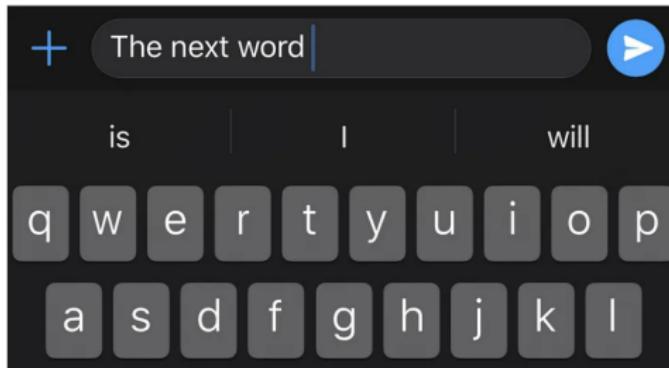


(Ref: Introduction to Generative AI - Google Cloud Tech)

Let's focus on the most popular modality ...

## What is a Language Models?

- ▶ While typing SMS, have you seen it suggests next word?
- ▶ While typing email, have you seen next few words are suggested?
- ▶ How does it suggest? (suggestions are not random, right?)
- ▶ In the past, for “Lets go for a . . .”, if you have typed ‘coffee’ 15 times, ‘movie’ say 4 times, then it learns that. Machine/Statistical Learning.
- ▶ Next time, when you type “Lets go for a ”, what will be suggested? why?
- ▶ This is called Language Model. Predicting the next word. When done continuously, one after other, it spits sentence, called Generative Model.



Next word prediction using language modeling in keyboards(Mandar Deshpande)

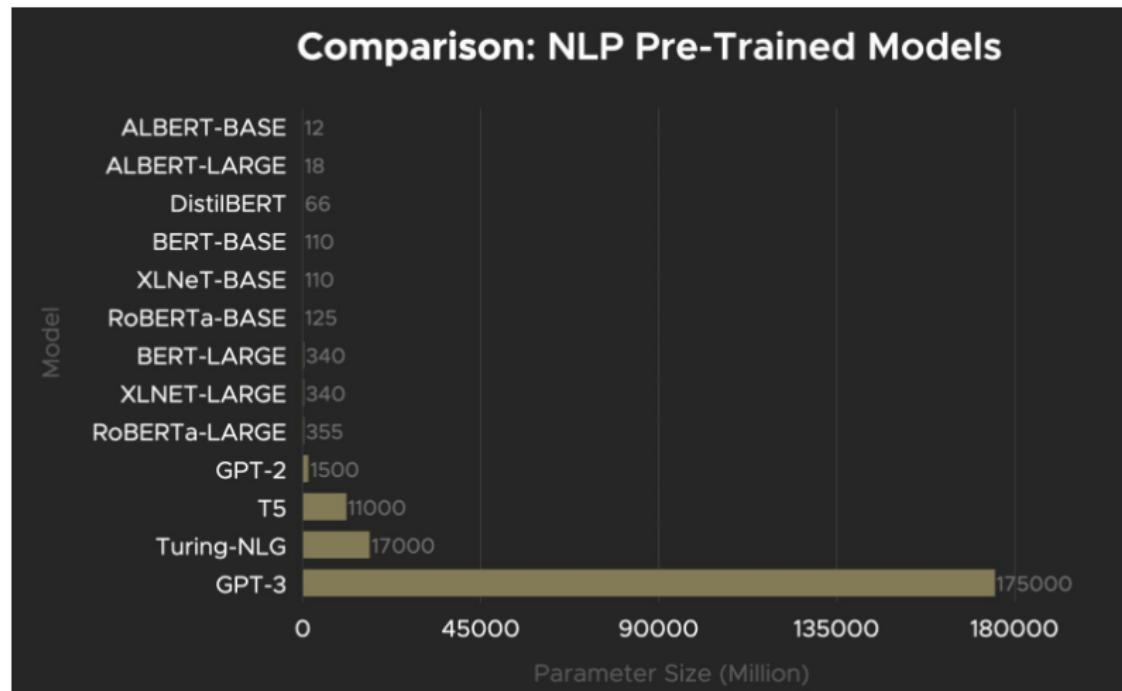
YHK

## Why they are called Large? Corpus

### GPT: Generative Pre-trained Transformers

- ▶ GPT-1 is pre-trained on the BooksCorpus dataset, containing 7000 books amounting to 5GB of data
- ▶ GPT-2 is pre-trained using the WebText dataset which is a more diverse set of internet data containing 8M documents for about 40 GB of data
- ▶ GPT-3 uses an expanded version of the WebText dataset, two internet-based books corpora that are not disclosed and the English-language Wikipedia which constituted 600 GB of data

## Why they are called Large? Parameters

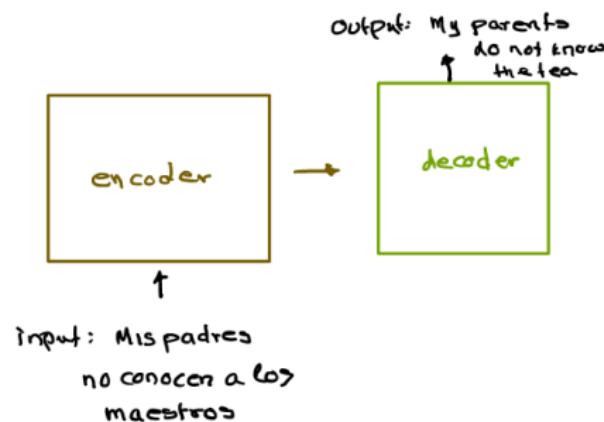


(Ref: Deus.ai <https://www.deus.ai/post/gpt-3-what-is-all-the-excitement-about>)

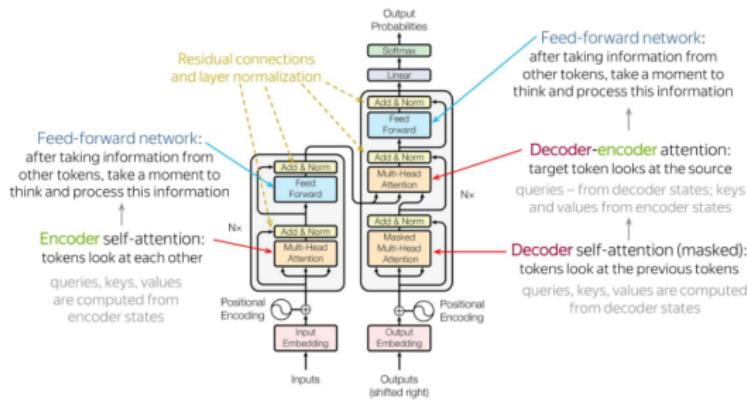
YHK

## Underlying Architecture: Transformers

- ▶ In its heart it contains an encoding component, a decoding component, and connections between them.
- ▶ The Transformer is a model that uses attention to boost the speed with which seq2seq with attention models can be trained.
- ▶ The biggest benefit, however, comes from how The Transformer lends itself to parallelization. How?



# Transformer Models

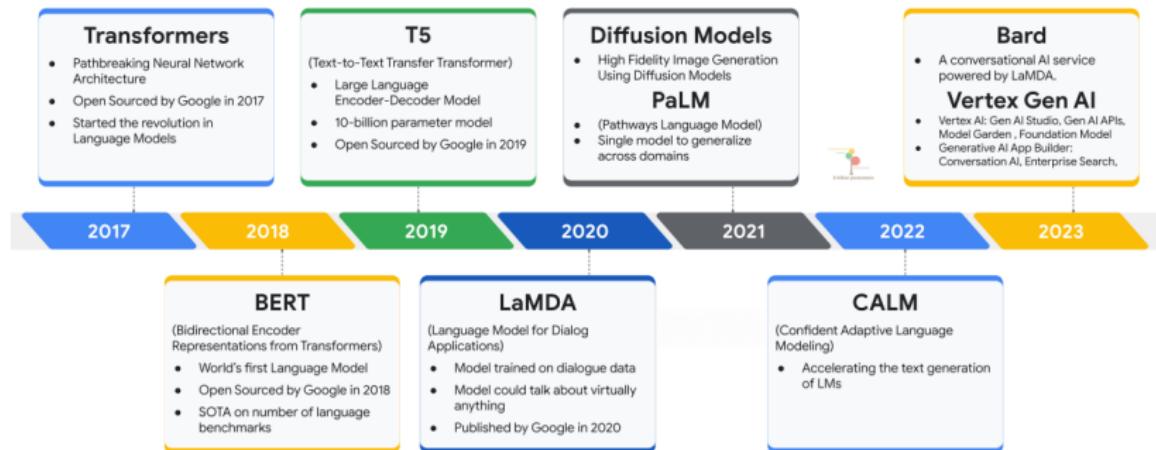


(Ref: The Complete Prompt Engineering for AI Bootcamp (2023))

- ▶ No recurrence, so parallelization possible
- ▶ Context information captured via attention and positional encodings
- ▶ Consists of stacks of layers with various sublayers

Transformers are basis of (the most) Large Language Models

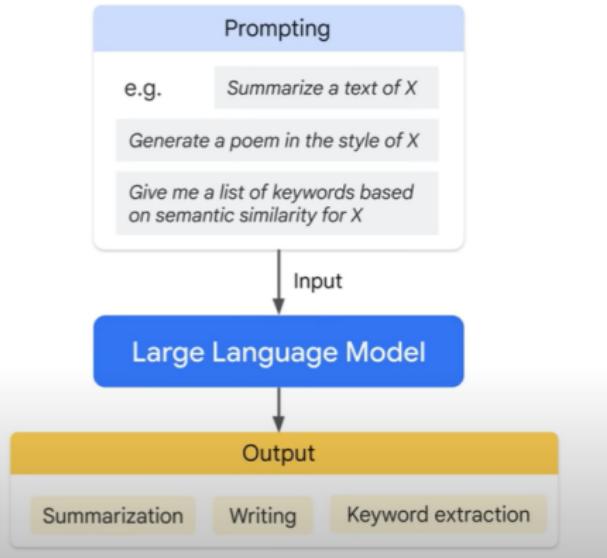
# The Progress of Models ...



(Ref: Primer on LLM and Gen AI - Google Cloud)

# Prompts driving Generative AI

**Prompt Design:**  
the quality of the  
input **determines** the  
quality of the output.



(Ref: Introduction to Generative AI - Google Cloud Tech)

What's IN these days . . .

YHK

# ChatGPT - A Tipping Point for Generative AI

- ▶ Released by OpenAI in November 2022
- ▶ Generative AI chatbot
- ▶ Rapid worldwide popularity
- ▶ 1 million users in 5 days
- ▶ Netflix took 3.5 years for same user count
- ▶ 100 million monthly active users by January 2023
- ▶ Fastest-growing application in history

# What's Inside?

## Technical Details

- ▶ Based on GPT3.5 Instruct architecture
- ▶ Estimated 175 billion parameters
- ▶ Fine-tuned on chat-specific task
- ▶ Curated dataset for fine-tuning

# Improving ChatGPT with RLHF

- ▶ Key technique: Reinforcement Learning from Human Feedback (RLHF)
- ▶ Trains language model to align with human preferences
- ▶ Collects human feedback on model-generated text
- ▶ Updates model's parameters using feedback
- ▶ Enhances ChatGPT responses' quality
- ▶ Increases factual, informative, and creative output

## Midjourney: Image Generation Model

- ▶ Developed by Midjourney Inc.
- ▶ Released in July 2022
- ▶ Architecture details undisclosed
- ▶ High-quality image generation
- ▶ Wide variety of styles and genres

# Meta Releases LLaMA

## Open Source LLMs Explode!

- ▶ February 2023: Meta releases LLM "LLaMA"
- ▶ LLaMA: 65-billion parameter model
- ▶ Trained on extensive text and code dataset

## Significance of LLaMA Release

- ▶ One of the largest public LLMs
- ▶ Suited for complex and challenging tasks
- ▶ Open source, initially for research purposes
- ▶ Model weights leaked online, accessible to all
- ▶ Sparked development of numerous open source LLMs



# Anthropic Claude

The screenshot shows a user interface for generating product names. At the top, there's a header with a square icon, three dots, and a search bar containing the text "Fit Shaker: Fast and Healthy". Below the search bar are two buttons: "Chat" and "Edit Chat". A purple sidebar on the right lists "Product names: HomeShaker, Fit Shaker, QuickShake, Shake Maker", "Product description: A pair of shoes that can fit any foot size.", and "Seed words: adaptable, fit, omni-fit". The main area displays a message: "Here are some suggested product names based on the seed words:" followed by a list of six items: "OmniFit", "AdaptiShoe", "UniSole", "One Size Fits All", and "FlexiFit". At the bottom, there's a text input field with placeholder text "Write a message..." and a blue send button with a white arrow.

- ▶ Uses Constitutional AI rather than RLHF
- ▶ Constitutional AI trains to follow a set of high-level principles or rules, such as a constitution, that specify the desired behavior and outcomes of the system.
- ▶ RLHF uses human feedback, such as ratings, preferences, or corrections, to optimize a language model or an agent's policy using reinforcement learning

(Ref: The Complete Prompt Engineering for AI Bootcamp (2023))

# Google's Latest LLM: Gemini and Gemma

## Google Releases Gemini

- ▶ December 2023: Rebranded from Bard to Gemini
- ▶ Advanced multimodal LLM for language, audio, code, and video
- ▶ Trained on extensive datasets for improved contextual understanding
- ▶ Real-time information access through Google Search integration

## Introducing Gemma

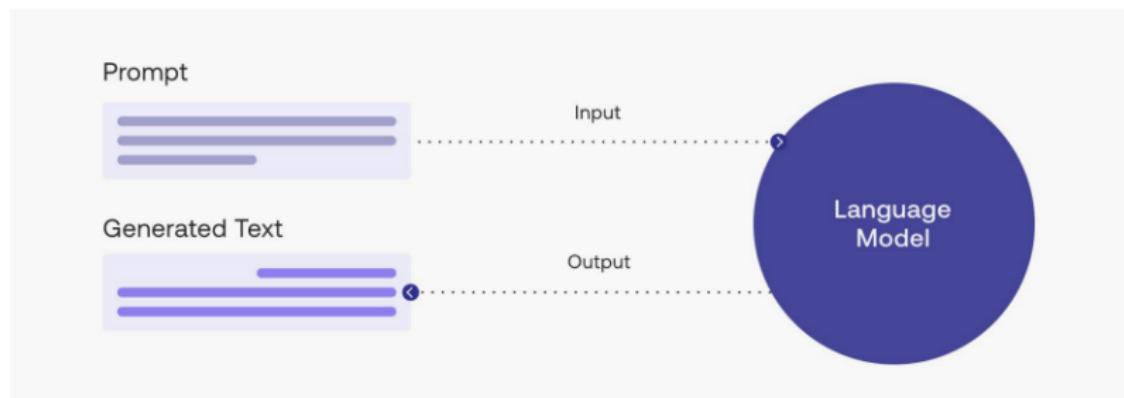
- ▶ Specialized version of Gemini for personal assistance, on Hugging Face
- ▶ Helps manage tasks, schedules, and reminders effectively
- ▶ Deep integration with Google Calendar and productivity tools

# Overview

YHK

# What is Prompt Engineering?

Prompt engineering is a NLP concept that involves discovering inputs that yield desirable or useful results



(Ref: Cohere <https://docs.cohere.ai/docs/prompt-engineering>)

# What is Prompt Engineering?

- ▶ For prompt What `is` 1,000,000 \* 9,000? GPT-3 (text-davinci-002) (sometimes) answers 9,000,000 (incorrect). This is where prompt engineering comes in.
- ▶ If, instead of asking What is 1,000,000 \* 9,000?, we ask What `is` 1,000,000 \* 9,000? Make sure to put the right amount of zeros, even `if` there are many:, GPT-3 will answer 9,000,000,000 (correct).
- ▶ Why is this the case? Why is the additional specification of the number of zeros necessary for the AI to get the right answer? How can we create prompts that yield optimal results on our task?
- ▶ That's Prompt Engineering.

(Ref: <https://learnprompting.org/docs/basics/prompting>)

## Sample (Simple) Gen AI Applications of Prompts

# Text Summarization

- ▶ Text summarization is a standard task in natural language generation.
- ▶ Summarizing information about antibiotics.

Prompt:

2 Antibiotics are a type of medication used to treat bacterial infections. They work by either killing the bacteria or preventing them from reproducing, allowing the body's immune system to fight off the infection. Antibiotics are usually taken orally in the form of pills, capsules, or liquid solutions, or sometimes administered intravenously. They are not effective against viral infections, and using them inappropriately can lead to antibiotic resistance.

Explain the above in one sentence:

4

Output:

6 Antibiotics are medications used to treat bacterial infections by either killing the bacteria or stopping them from reproducing, but they are not effective against viruses and overuse can lead to antibiotic resistance.

# Information Extraction

- ▶ Language models can perform classification and other NLP tasks.
- ▶ Example: Extracting specific information from a paragraph.

Prompt:

2 Author-contribution statements and acknowledgements in research papers should state clearly and specifically whether, and to what extent, the authors used AI technologies such as ChatGPT in the preparation of their manuscript and analysis. They should also indicate which LLMs were used. This will alert editors and reviewers to scrutinize manuscripts more carefully for potential biases, inaccuracies and improper source crediting. Likewise, scientific journals should be transparent about their use of LLMs, for example when selecting submitted manuscripts.

4 Mention the large language model based product mentioned in the paragraph above:

6 Output:

The large language model based product mentioned in the paragraph above is ChatGPT.

# Question Answering

- ▶ Structured prompts combine instructions, context, input, and output indicators.
- ▶ Example: Using context to answer a specific question. Context obtained from [Nature](<https://www.nature.com/articles/d41586-023-00400-x>).

1 Answer the question based on the context below. Keep the answer short. Respond "Unsure about answer" if not sure about the answer.

3 Context: Teplizumab traces its roots to a New Jersey drug company called Ortho Pharmaceutical. There, scientists generated an early version of the antibody, dubbed OKT3. Originally sourced from mice, the molecule was able to bind to the surface of T cells and limit their cell-killing potential. In 1986, it was approved to help prevent organ rejection after kidney transplants, making it the first therapeutic antibody allowed for human use.

5 Question: What was OKT3 originally sourced from?

7 Answer:

9 Output:

Mice.

YHK

# Text Classification

- ▶ Providing examples in prompts improves model specificity.
- ▶ Simple text classification.

```
Prompt:  
2 Classify the text into neutral, negative or positive.  
Text: I think the food was okay.  
4 Sentiment:  
Output:  
6 Neutral
```

# Text Classification

Example: Using examples to enforce specific behavior.

```
Prompt:  
2 Classify the text into neutral, negative or positive.  
Text: I think the vacation is okay.  
4 Sentiment: neutral  
  
6 Text: I think the food was okay.  
Sentiment:  
8 Output:  
neutral
```

# Conversation

- ▶ Instructing system behavior through role prompting.
- ▶ Example 1: Creating a technical and scientific AI research assistant.

```
1 Prompt:  
The following is a conversation with an AI research assistant. The assistant  
tone is technical and scientific.  
3  
Human: Hello, who are you?  
5 AI: Greeting! I am an AI research assistant. How can I help you today?  
Human: Can you tell me about the creation of blackholes?  
7 AI:  
Output:  
9 Sure! Black holes are regions of spacetime where the gravitational force is so  
strong that nothing, not even light, can escape from it. They are  
created when a very massive star dies and its core collapses in on  
itself, forming a singularity of infinite density. The intense gravity  
of the singularity pulls in all the matter and radiation around it,  
creating the black hole.
```

## Code Generation

- ▶ Language models are effective for code-generation tasks.
- ▶ Example 1: Generating a program to greet the user.

```
1 Prompt:  
/*  
3 Ask the user for their name and say "Hello"  
*/  
5 Output:  
let name = prompt("What is your name?");  
7 console.log('Hello, ${name}!');
```

## Code Generation

- ▶ Example 2: Generating MySQL queries from a schema description.

```
1 Prompt:  
2 """  
3 Table departments, columns = [DepartmentId, DepartmentName]  
Table students, columns = [DepartmentId, StudentId, StudentName]  
5 Create a MySQL query for all students in the Computer Science Department  
6 """  
7 Output:  
8 SELECT StudentId, StudentName  
9 FROM students  
WHERE DepartmentId IN (SELECT DepartmentId FROM departments WHERE  
    DepartmentName = 'Computer Science');
```

# Reasoning

- ▶ Language models require advanced techniques for reasoning tasks.
- ▶ Example 1: Simple arithmetic.

```
Prompt:  
2 What is 9,000 * 9,000?  
Output:  
4 81,000,000
```

# Reasoning

- ▶ Example 2: Problem-solving with step-by-step instructions.

Prompt:

2 The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.

Solve by breaking the problem into steps. First, identify the odd numbers, add them, and indicate whether the result is odd or even.

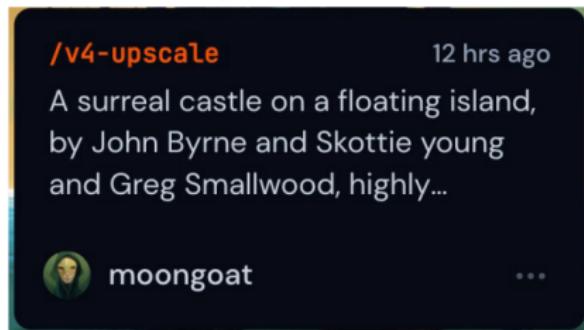
4 Output:

Odd numbers: 15, 5, 13, 7, 1

6 Sum: 41

41 is an odd number.

# Image Generation



(Ref: Prompt Engineering Sudalai Rajkumar)

Models / Tools: Dall-E , Midjourney, Stable Diffusion

# Programmatic Calling of Prompt

```
1 import openai
2 import os
3
4 from dotenv import load_dotenv, find_dotenv
5 _ = load_dotenv(find_dotenv())
6
7 openai.api_key = os.getenv('OPENAI_API_KEY') # for langchain it does it
8     automatically
9
10 def get_completion(prompt, model="gpt-3.5-turbo"):
11     messages = [{"role": "user", "content": prompt}]
12     response = openai.ChatCompletion.create(
13         model=model,
14         messages=messages,
15         temperature=0, # this is the degree of randomness of the model's output
16     )
17     return response.choices[0].message["content"]
```

# Making of a Sandwich

YHK

## Basic

Explain how to make a peanut butter and jelly sandwich

YHK

## Adding Roles

As a chef, explain to your assistant how to make a peanut butter and jelly sandwich



## Adding Constraints

Make a nut-free version of the sandwich due to a customer's nut allergy

## Adding Examples

Create two unique variations of the classic sandwich. Banana Nut Crunch: . . .

Triple Berry Blast: . . .



## Adding Contextual Information

As the head chef at 'The Sandwich Haven,' guide your new assistant to create specials for the menu

## Incorporating Feedback

Improve the sandwich based on customer feedback for less sweetness and a creative twist

## Time Constraints and Prioritization

Prepare an alternative fruit version for testing within a tight deadline

## Incorporating Multidisciplinary Knowledge

Use food presentation and garnishing techniques for a visually appealing sandwich

## Addressing Dietary Preferences

Prepare a vegan version using plant-based alternatives for all ingredients

## Reflection and Iteration

Reflect on feedback and iteratively refine the sandwich for better taste and appeal

## Self-Criticism

Explain how to make a peanut butter and jelly sandwich. Please re-read your above response. Any mistakes? If so, please identify and make the necessary edits.

## Chain-of-Thought

Explain how to make a peanut butter and jelly sandwich. Let's think step by step.

## Self-Consistency

Here are recipes of multiple sandwiches. Sandwich 1: recipe 1. Sandwich 2: recipe 2. .... Explain how to make a peanut butter jelly sandwich.



# Conclusions

YHK

# Progression

Models for prediction:

- ▶ On data, derive features, put statistical techniques like regression. One model per task. That's Machine Learning.
- ▶ Feed raw data, employ neural networks. One model per task. That's Deep Learning.
- ▶ Use Text data, get embeddings, use ML/DL, say for classification. One model per task. That's Natural Language Processing.
- ▶ Train neural network on large corpus, store weights and architecture, then add final layers for say classification on custom data+labels. That's Pretrained model. One model, many tasks.
- ▶ Train Large Language Model, just supply instructions on what to do, works. One model many tasks. Zero-shot, few-shots.

(More info at SaaS LLM <https://medium.com/google-developer-experts/saasgpt-84ba80265d0f>)

# New Programming Language?



Andrej Karpathy ✅  
@karpathy

...

The hottest new programming language is English

1:44 AM · Jan 25, 2023 · 1.9M Views

---

2,050 Retweets 284 Quote Tweets 17.9K Likes

---

(Ref: Prompt Engineering Sudalai Rajkumar)

# Summary of Prompt Engineering

## ► **Definition of Prompts**

- ▶ Prompts are initial text inputs provided to a model.
- ▶ Used by the model to generate responses or accomplish tasks.

## ► **Role of Prompts**

- ▶ Sets of instructions for AI or chatbots (e.g., ChatGPT).
- ▶ Applied in various tasks, including summarization, arithmetic problem-solving, and question-answering.

## ► **Objective of Prompt Engineering**

- ▶ Goal: Refine prompts to enhance model accuracy and relevance in outputs.
- ▶ Central to improving the performance of language models.

## ► **Prevalent Prompt Types**

- ▶ Various prompt types exist, with a focus on two widely used methodologies:
  - ▶ Zero-shot prompting
  - ▶ Few-shot prompting

# ChatGPT Ultimate Prompting Guide

- ▶ Tone: Specify the desired tone (e.g., formal, casual, informative, persuasive).
- ▶ Format: Define the format or structure (e.g., essay, bullet points, outline, dialogue).
- ▶ Act as: Indicate a role or perspective to adopt (e.g., expert, critic, enthusiast).
- ▶ Objective: State the goal or purpose of the response (e.g., inform, persuade, entertain).
- ▶ Context: Provide background information, data, or context for accurate content generation.
- ▶ Scope: Define the scope or range of the topic.
- ▶ Keywords: List important keywords or phrases to be included.
- ▶ Limitations: Specify constraints, such as word or character count.
- ▶ Examples: Provide examples of desired style, structure, or content.
- ▶ Deadline: Mention deadlines or time frames for time-sensitive responses.

(Ref: LinkedIn post by Generative AI, Twitter by Aadit Sheth, Source : Reddit)



# ChatGPT Ultimate Prompting Guide

- ▶ Audience: Specify the target audience for tailored content.
- ▶ Language: Indicate the language for the response, if different from the prompt.
- ▶ Citations: Request inclusion of citations or sources to support information.
- ▶ Points of view: Ask the AI to consider multiple perspectives or opinions.
- ▶ Counter arguments: Request addressing potential counterarguments.
- ▶ Terminology: Specify industry-specific or technical terms to use or avoid.
- ▶ Analogies: Ask the AI to use analogies or examples to clarify concepts.
- ▶ Quotes: Request inclusion of relevant quotes or statements from experts.
- ▶ Statistics: Encourage the use of statistics or data to support claims.
- ▶ Visual elements: Inquire about including charts, graphs, or images.
- ▶ Call to action: Request a clear call to action or next steps.
- ▶ Sensitivity: Mention sensitive topics or issues to be handled with care or avoided.

(Ref: LinkedIn post by Generative AI, Twitter by Aadit Sheth, Source : Reddit)



## Interaction Guidelines: Avoid Misuses

- ▶ Factual Accuracy: Interactions must be free from factual inaccuracies that can be challenged by social media or journalists.
- ▶ Negative Debates: Avoid discussing topics that fuel negative or concerning online debates, such as AI sentience, AI in education, AI-driven job displacements, and politically divisive issues.
- ▶ Minors' Involvement: Do not include use cases specifically targeting or involving individuals under 18 years old.
- ▶ Sensitivity and Misinformation: Prevent the inclusion of sensitive, misleading, or hazardous responses.
- ▶ Search and Google Assistant: Interactions that require basic, straightforward answers are better suited for Search or Google Assistant.
- ▶ Financial/Legal/Medical Advice: Refrain from providing advice related to financial matters, legal issues, or medical concerns.
- ▶ Brand Names and Trademarks: Avoid mentioning specific brand names, trademarks, or public figures (except historical figures).
- ▶ No Reviews or Tweets: Do not request reviews of restaurants, businesses, or tweets to minimize the risk of associating with bots.
- ▶ Avoid Personification: Refrain from personifying the product or brand and from encouraging users to address Bard by name.



## Limitations

Boie is a real company, the product name is not real. So, see what you get ...

```
prompt = f"""
2 Tell me about AeroGlide UltraSlim Smart Toothbrush by Boie
"""
4 response = get_completion(prompt)
print(response)
```

# What Next?

YHK

# The Career of the Future

## Software 3.0

---

“Programming [is] moving from curating datasets to curating prompts to make the meta learner “get” the task it’s supposed to be doing.”

Source: [@karpathy](#)



(Ref: The Complete Prompt Engineering for AI Bootcamp (2023))

## New Roles?

Coming up with good prompt is a combination of art and science



Alexandr Wang ✅  
@alexandr\_wang

...

Today, [@goodside](#) joined [@scale\\_AI](#) as a Staff Prompt Engineer.

I am going to assert that Riley is the first Staff Prompt Engineer hired \*anywhere\*.

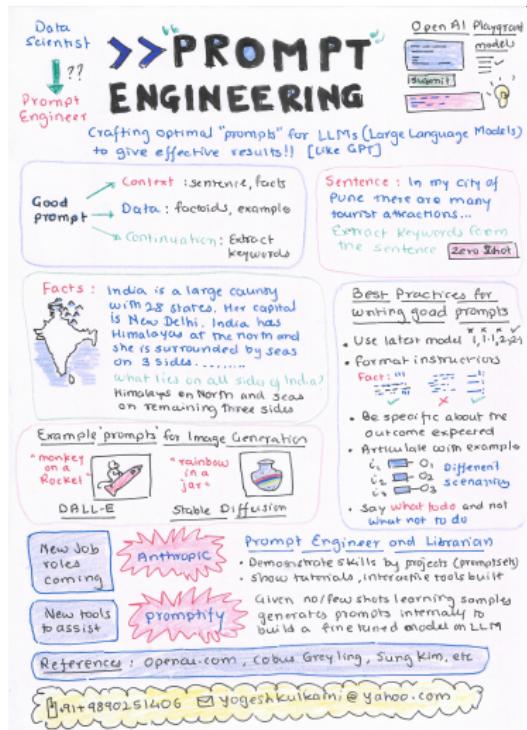
(Ref: Prompt Engineering Sudalai Rajkumar)

## Read on to learn how to engineer good prompts!

- ▶ Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).  
<https://doi.org/10.18653/v1/2020.emnlp-main.346>
- ▶ Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners.
- ▶ Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2022). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys.  
<https://doi.org/10.1145/3560815>
- ▶ Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners.
- ▶ Zhao, T. Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). Calibrate Before Use: Improving Few-Shot Performance of Language Models.



# My Sketchnote



(Ref: <https://medium.com/technology-hits/prompting-is-all-you-need-5dddb82bd022>)

## Take Aways

Prompt Engineering is an Iterative Process:

- ▶ Try something
- ▶ Analyze where the results do not match the expectations
- ▶ Clarify instructions, gives examples, specify output format, specify constraints, etc
- ▶ Test on a batch of known results.

## Quality of Ideas

"I don't think we'll still be doing prompt engineering in five years [i.e.] figuring out how to hack the prompt by adding one **magic word** to the end that changes everything else."

"What will always matter is the **quality of ideas** and the understanding of what you want."

Source: [Sam Altman](#)

(Ref: The Complete Prompt Engineering for AI Bootcamp (2023))

## Resources

- ▶ Prompt Engineering Guide  
<https://github.com/dair-ai/Prompt-Engineering-Guide>
- ▶ Awesome ChatGPT Prompts  
<https://github.com/f/awesome-chatgpt-prompts/>
- ▶ ChatGPT Prompt Engineering for Developers - Deep Learning AI
- ▶ Learn Prompting <https://learnprompting.org/docs/intro>
- ▶ Types of Prompts with Practical examples - Dr. Naveed Siddiqui
- ▶ AI Prompt Database  
<https://justunderstandingdata.notion.site/d98dcc9a6736471584d53cc8b2a5c30d?v>

Thanks ...

- ▶ Search "**Yogesh Haribhau Kulkarni**" on Google and follow me on LinkedIn and Medium
- ▶ Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.
- ▶ Email: yogeshkulkarni at yahoo dot com

(<https://www.linkedin.com/in/yogeshkulkarni/>, QR by Hugging Face

QR-code-AI-art-generator, with prompt as "Follow me")

