Outline

→ What is Gen AI

→ Examples — AI application

→ End-to-End Gen AI application — Motivation

→ Focus → Gen AI
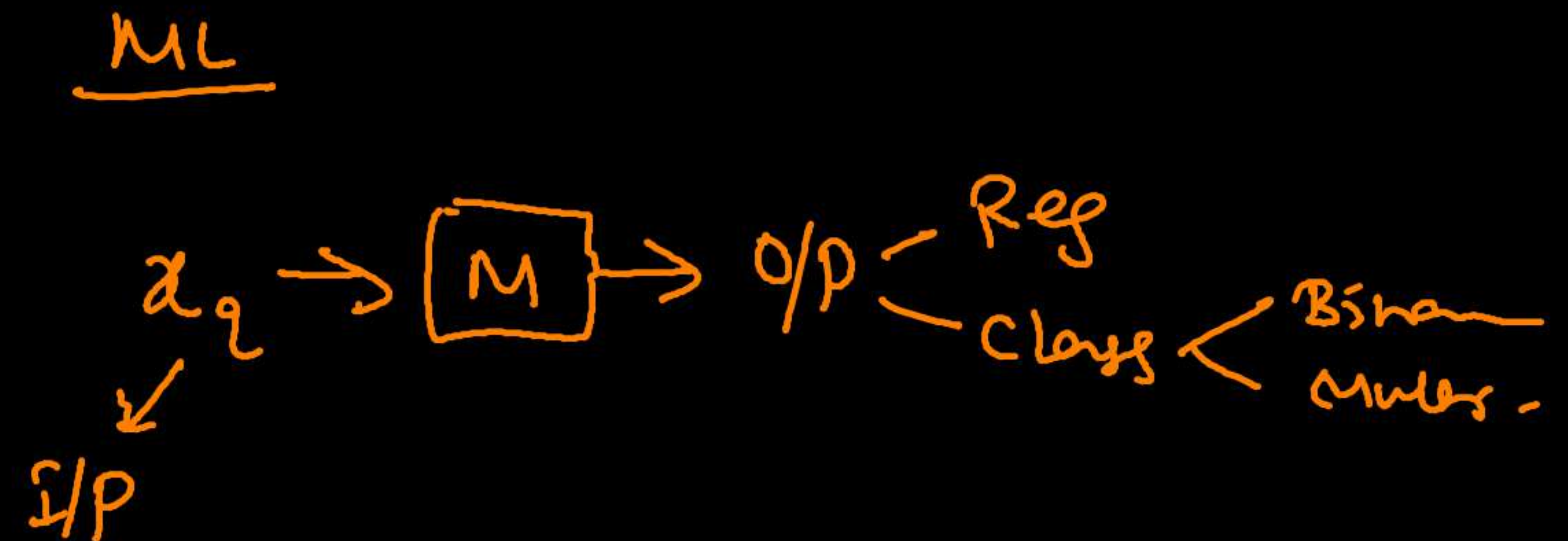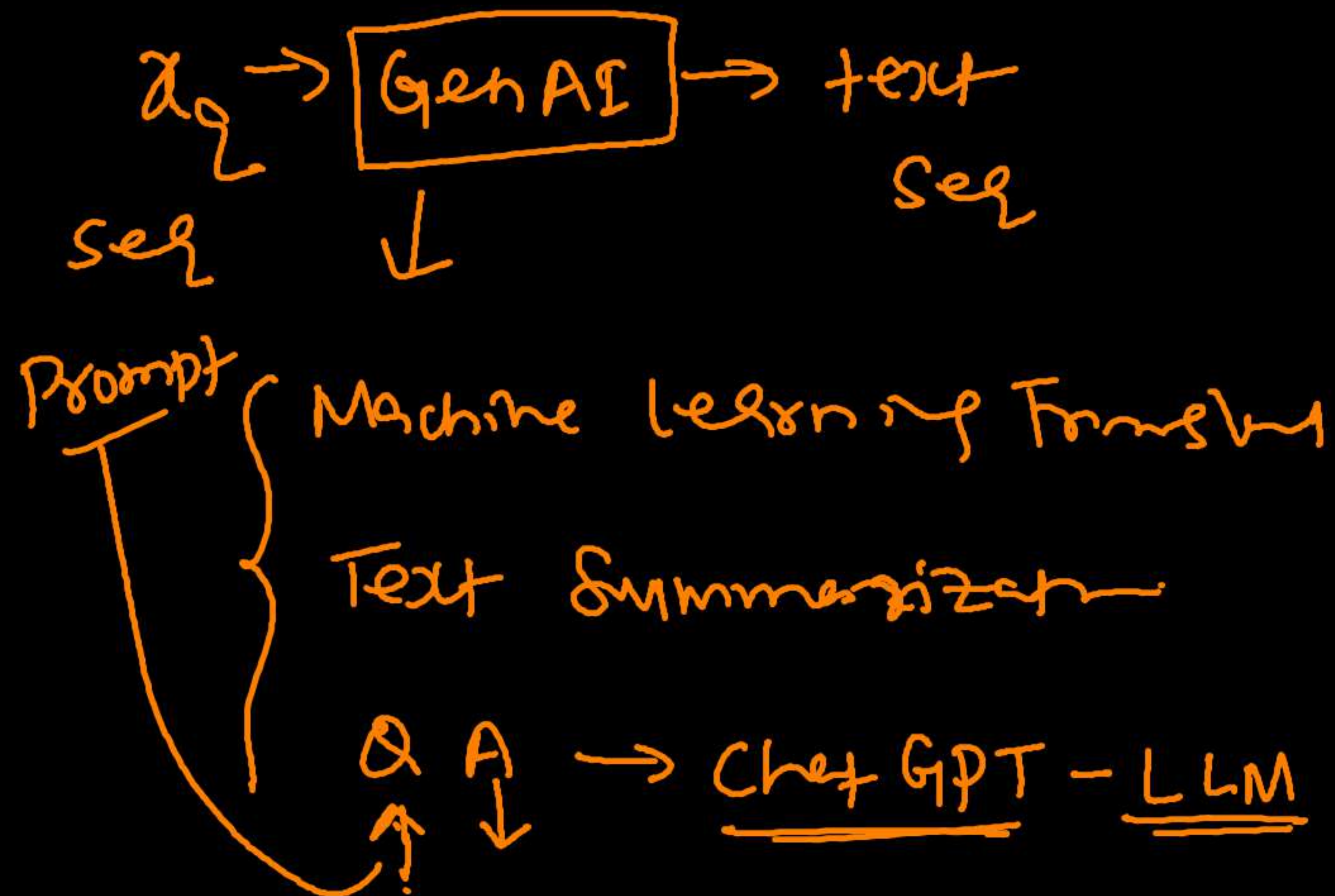
GPT

ML+DL

→ Prompt Engineering + API
→ RAG + Langchain
→ Fine Tuning
→ Stable diffusion → Image
→ LLMops

Text

Building Application Powered By Gen AI

① Prompt engineering

$x_q \rightarrow$ **GenAI** $\rightarrow$ text
seq

seq $\downarrow$

Prompt {
  Machine learning Translat

  Text Summarization

  Q A $\rightarrow$ Chat GPT - LLM
}

ML

$x_q \rightarrow$ **M** $\rightarrow$ O/p < Reg
         Class < Binar
                Muler

x_q ↓
I/P

2000 - 2017 — <u>DL</u>

2014 — Encoder - Decoder

2015 — Attention mechanism

2017 — Attention is all you need — Transformers

Parallel

Encoder   Decoder
&
self attention.

MLM
NWP
2018 — BERT — DistilBERT, MobileBERT
TinyBERT, RoBERTABERT

2018 — VMLFit ⟹ Language model
GPT 1

Q:- How do you build a LLM?
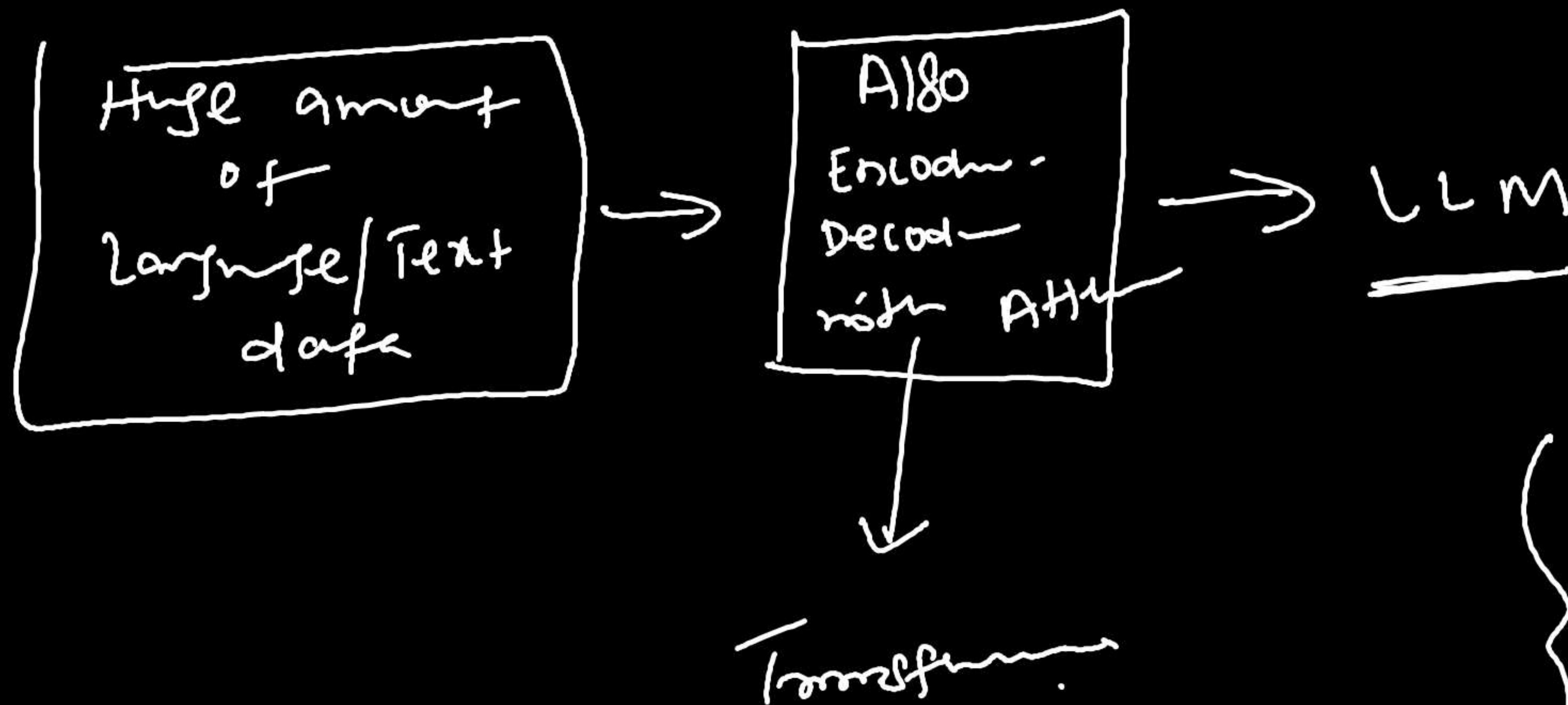
Open AI

GPT 3.5 | 4 | 40

Meta - LLama

Google AI
　└→ Gemini

| Huge amount of Language/Text data | → | Algo Encoder- Decod- with Atte | → | L L M |

　　　　　　　　↓

　　　　Transform-

Lots of money - Inv-

Data Comput-

Time & R-

Q:- is this way to build?

End-

* Better approach → use already existing LLM's which
  are pre-trained on language task.

  ↓

  use them to solve Downstream task

  { Machine Translat____ Approach
    Text Sum____   ① Prompt Engineering
    Q A Syst____   ② RAG
    chatbot   ③ fine Tuning .

$\Rightarrow$ Design better input to get good quality result from a (LLM)

Step 1 :- Identify a good large language model

$\begin{cases} \underline{GPT} \text{ from OpenAI} \\ \underline{Gemini} \text{ from Google} \\ \underline{LLAMA} \text{ || meta} \end{cases}$ $\underline{SOTA}$

Step 2 :- Experiment with the following ——— LLM
(Prompt en)

$\begin{cases} \textcircled{1} \quad \text{Prompt Engineering} \\ \textcircled{2} \quad RAG \\ \textcircled{3} \quad \text{Fine Tuning} \end{cases}$

RAG
LLM

Fim LLM

Q: What is a Prompt?

A: Input to an LLM.

Prompt engineering / Design

OpenAI-ky = "key"

Client = OpenAI(apikey= OL)

① Write clear instruction

② Few/Zero Shot Learning

③ Output Structure

④ Prompting Personas

⑤ Give the model time to think