Audio
Video
Participants 29
Chat
Share
Pause
Annotate
Show webinar
More
Talking: Learnvista Private Lim...
You are screen sharing
Stop share
Sanjoy raised hand
View

Gen AI

Intro to LangChain Model (LLM & Chat Models)

→ ML → NLP & Text Pre-Processing - BOW/TF-IDF, W2V, n-grams, glove...

→ Gen AI → RNN/LSTM/GRU

Seq 2 seq + Encoder-Decoder + Attention Machanism + Transformers

↓ Parallelism

↓

prompt engineering → OpenAI → GPT, → DALLE, → wishper } API

→ Limited knowledge
→ Hallucination

RAG

Google AI { Gemini 1.5 - Flash } - API

Llama 2 — Hugging Face — Transformers | Acceler
8 bit, pipeline

Audio
Video
Participants 33
Chat
Share
Pause
Annotate
Show webinar
More
Talking: Learnvista Private Lim...
You are screen sharing
Stop share
Sourav Karmakar raised hand
View

CRED – Application

↳ To pay credit bill, vehicle Fastag – in—

electrict.
gas bill

Credit card

CC1 – HDFC → 5K

CC2 → HSBC → 2K

CC3 → ICICI → 3K

CRED unifies the credit card bill

Cashback & Reward point

Pain point

Langchain is a way to UNIFY GenAI APP

App powered by GenAI

→ Gemini
→ OpenAI GPT_4o mini
→ Hugging face → meta AI - LLama 3.1

OpenAPI — code implemented is very different
or
Google API

Building Apps powered by GenAI using OpenAI

Audio
Video
26
Participants
30
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share
Rohit Vishwakarma raised hand
View
client = OpenAI (API_key)
Request
Raw input
client.chat.completions.
create (model = ' ')
messages =
parameters
Prompt
model = GPT 3.5
4o
4omini
n
temp
top-p
top-k
Response
response.
choices[0].
message.content
Output
App

Audio
Video
26
Participants
30
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share
View
×
Bulding Apps powered by GenAI using Google AI
genai.configure(api_key
Request
genai.GenerativeModel(model,
System
Model
Prompt
Gensession.
top_k
top_p
temp,
candidate
count
Response.text
output
Respon

Rohit Vishwakarma raised hand

Audio
Video
26
Participants
30
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share
Rohit Vishwakarma raised hand
View
×
Gen AI Application
convert Raw user I/p to prompt
Raw I/p (Request)
Logic 1 .
Prompt
①
Model
Logic2
output
Respons.

Langchain

① Model

② Templete - Logic 1 - take the user input & convert into a proper prompt. output structure

③ Output parser (logic 2) → how to see your output

④ Chain

Chain

Templte | Model GPT | output = Res

Audio
Video
36
Participants
1
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share

1. MODEL < LLM → completion X -

Chat LLM → conversational → Good for Conversational BOT

langchain-OpenAI < LLM - Instruct based

Chat LLM → Conver Chat

OpenAI   Chat OpenAI

langchain-google-genai — Google GenerativeAI

Chat Google Generative AI

langchain-Cohere < Cohere

Chat Cohere

langchain-anthropic < Antropic - LLM — Good for instruction

Chat Anthropic - Chat LLM — Good for Conv

Audio
Video
Participants
36
Chat
3
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share
Talking: Learnvista Private Lim...

(before 2024 march) old model

LLM → instruction based model

~~Open~~ AI

Chat OpenAI ← Chat Based model ✓

Open AI → from langchain_openai import

Chatmodel = ChatOpenAI(- - —)

Google AI — " " - google-genai " Chat Google Generative AI

Chatmodel = Chat Google Generative AI( )

Anthropic = " " - anthropic " Chat Anthropic

Chatmodel = Chat Anthropic( )

Cohere = " " - cohere " Chat-cohere

Chatmodel = ChatCohere( ))

- API Key
- Name of Chat LLM
- Parameter

Chatmodel.invoke(user_prompt) ✓

Talking: Learnvista Private Lim
Audio
Video
35
Participants
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share

Langchain

① Model — LLM & ChatLLM
X ✓

② prompt template

③ output structuring

④ Chain

⑤ Memory → Conversational Bot

⑥ vector DB's — Chrome DB | BERT | GPT

Audio
Video
Participants 34
Chat 1
Share
Pause
Annotate
Show webinar
More
e Lim...
You are screen sharing
Stop share
Switch kernel
Req
Response
APP UI
Template
M
OpenAI
GoogleAI
Prompt Templet
LLM
O/P

design Template for prompt

→ Design Output parser for the model output

→ my need here is

model Agnostic (independent)

Logic 1 – Template ~

Logic 2 = Output P— -

Audio
Video
33
Participants
Chat
Share
Pause
Annotate
Show webinar
More
e Lim...
You are screen sharing
Stop share

eg ② => Explain ____ topic in datasci-

↑

outlier

Templates

Prompt Template

ChatPrompt Template

String - LLm

list of Chat MSS - Chat model

Raw
I/p
Template
Prompt
Model

① Tell me a ____ Joke abut ____

↑ funny

↑ DataScien: -

Audio
Video
25
Participants
Chat
Share
Pause
Annotate
Show webinar
More
You are screen sharing
Stop share

adj

Cont—

PromptTemplate.from_template (Tell me a ___ Joke about ___)

Strings

.format()

Tell me a funny Joke about Data Science.

.format_prompt()

HumanMessage(- -'

Tell me a funy Joke about data Sci))

Human

System

ai

→ c.

Audio
Video
33
Participants
7
Chat
Share
Pause
Annotate
Show webinar
More
e Lim...
Stop share
View

You are screen sharing

Atriyo Daripa raised hand

```
topic = input("Enter a topic")

template =
```