LANGCHAIN is a rimplified
framework that a implifies the process
of building LIM powered Apps
-) tools 72
-7 Abstration S
Billing and han in the head to fix which
-> Building our LLM App isn't straight-forwire
Data
UNEY -> LLM - History
2 works API cally
Other Google LLM Search
-> To build an intelligent Applike above, you will
need Lang Chain.
-> Long Chain has name interface across
all LLM rendors.
-> Long Cheun (LC) have prompt management
, optimisation, residention
7 LC also par document lo soders (.pdf,
word, emails, etc)
-> LL supports tool calling, etc
-> LC creates abstraction
"chains"-
-> Lang Chain API
Long Chain
$\sqrt{\sqrt{2}}$
Lorg (han commity)
La Laglan Core
Long Chain Model IO
Log Choin Model IO  Expresson Legeuge Document Load  Wester Stars
\ \(\mathcal{E} \) \(\m
Agent too by
-> Long Cherin Hub
Contributed by Community.
contributed by community.
Python 3.11.9
( Lay chown
piperu (ang chain - openai
piperis  [ang chain - openai  install  lang chain - commutes
piperil langchain-openai install langchain-commuty hangchainhub
piperis  [ang chain - openai  install  lang chain - commutes
pipend langchain-openai install langchain-commuty hangcheinhub black
piperny langchain-openai install langchain-commuty hagcheinhub black  We used langchain core reman
pipend langchain-openai install langchain-commuty hangcheinhub black

Also install python-dotent

fromp Template:
Reampt is input to UM (instructions
tdet2)
PromptTemplate helps to translates
wer inputs, parameters into instrution
for LLM
Dictionary (Key: Value)
Prompt Template
Prompt Volue
] ilb to TTW: drouged broubt
LLM
C> 1 40 .
Chat Models:
Pronde avers to LLM va
Chatmodely
Listof Messages 11P
Chatmada
J. Charles a
Musage OIP
> The sales a post
→ They also support:
-7 Strubored Off
-> Muterned at ipp off-
-> By Combing prompt templates,
chatmodely, we can passitate to
chatmodels, we can passitate to
Chatmodels, we can passitate to
Chat modely, we can passitate to  LLMS.  -> We can combine this mith enternal APIS;
chat modely, we can passitate to  LLMS.  The writers at a set a polyment of the polyment of th
chatmodoly, we can passitate to  LLMS.  The writers, etc  in Long Chain to properly make an  Applicate
Chatmodoly, we can passitate to  LLMS.  The writers and APIS;  pdf readers, the writers, setc  in Long Chain to properly make an  Applicate
Chatmodely, we can passitate to  LLMS.  The writers, etc  in Long Chain to properly make an  Applicate  The contine / Chevin all there together  to make an Application
Chatmodoly, we can passitate to  LLMS.  The water and APIS;  pdf readers, the waters, etc  in Long Chain to proporty make an  Applicate  To we can combine / Chain all there together  to make an Application  Agent: In LLM world, an agent
Chatmodoly, we can possitate to  LLMS:  The waters and APIS;  pdf readers, the waters, etc  in Long Chain to properly make an Applicate  The control of their all there togethere  to make an Application  Agent: In LLM world, an agent  is a loop that repetedly reads a
chatmoddy, we can passitate to  LLMS.  The water and APIS;  pdf readers, the waters, setce  in Long Chain to properly make an  Applicate  The consideral Chain all their together  to make an Application  Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what actor
chatmoddy, we can passelata to  LLMS:  -7 We can combine this with external APIs;  pdf readers, the writers, setc  in Long Chain to properly make an  Applicate  -> We can combine / Chain all there together  to make an Application  -> Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what action  or tool all to make, feeds the result
chatmodely, we can passitate to  LLMS.  -> We can combine this with external HPIs;  pdf readers, the writers, etc  in Long Chain to properly make an  Applicate  -> We can combine / Chein all there together  to make an Application  -> Agent: In LLM world, an agent  is a loop that repetitedly reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the good is
chatmoddy, we can passelata to  LLMS:  -7 We can combine this with external APIs;  pdf readers, the writers, setc  in Long Chain to properly make an  Applicate  -> We can combine / Chain all there together  to make an Application  -> Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what action  or tool all to make, feeds the result
chatmoddy, we can passitate to  LLMS.  -7 We can combine this with external APIs;  pdf readers, the writers, etc  in Long Chain to properly make an  Applicate  -> We can combine / Chein all there together  to make an Application  -> Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the goal is  met
Chatrooddy, we can passidate to  LLMS:  The writers, after writers, etc.  In Long Chain to properly make an Applicate  The consine / Chain all their together to make an Application  Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what action or tool all to make, feeds the result back into the model until the goal is met  The Lamp Chain is a python framework.
Chatroodely, we can passidate to  LLMS:  The water and the waters, setce  in Long Chain to properly make an  Applicate  The con combine / Chain all their together  to make an Application  Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the goal is  met  > Lang Chain is a Python framework  that gives us buildry blocks
chatmoddy, we can passitate to  LLMS:  The waters the waters, etc  in Long Chain to properly make an  Applicate  To make an Application  Agent: In LLM world, an agent  is a loop that repetedly reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the goal is  met  > Lang Chain is a Python framework  that gives us buildry blocks  (prompt templates, Chains, mamories,
Chatroodely, we can passidate to  LLMS:  The water and the waters, setce  in Long Chain to properly make an  Applicate  The con combine / Chain all their together  to make an Application  Agent: In LLM world, an agent  is a loop that repeatedly reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the goal is  met  > Lang Chain is a Python framework  that gives us buildry blocks
chatmodely, we can peopleton to  LLMs.  The waters of the waters, setce in Long Chain to properly make an  Application  We can combine / Chein all their together to make an Application  Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the good is met  Lang Chain is a Python framown that gives us buildry hlocks (prompt templates, Chains, mamories, callbacks, tool wrappers) to build LLM appo
chatmoddy, we can passitate to  LLMS:  The concernation thus with external APIS;  pdf readers, the writers, etc  in Long Chown to proporty make an  Application  The concernation  Agent: In LLM world, an agent  is a loop that represently reads a  conversation, deades what action  or tool all to make, feeds the result  back into the model until the goal is  met  That gives us buildy blocks  (prompt templates, chains, memories,  allbacks, tool wrappend to build LLM apper  TREACT -> It is a prompting
Chatmoddy, we can passitate to  LLMS:  The water combine this with enternal APIS;  pdf readers, the writers, etc.  in Long Chain to properly make an Application  The make an Application  Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, decides what actor or tool all tomake, feeds the result back into the model until the good is met  That gives us buildry blocks (prompt templates, chains, memories, callbacks, tool wrappers) to brild LLM apple.  TEA (T -> It is a prompting pattern where the model (LLM)
Chatmodoly, we can passitate to  LLMS:  The can combine this with external APIS;  pot readers, the writers, etc  in Long Chown to proporty make an  Application  The can combine / Chown all their together  to make an Application  Agent: In LLM world, an agent  is a loop trad repeatedly reads a  conversation, decides what action  or tool all to make, feeds the result  back into the model until the goal is  met  That gives us buildry blocks  (prompt templates, chains, mamories,  callbacks, tool wrappers) to build LLM apple  TREACT -> It is a prompting  pattern where the model (LLM)  emits an interleaved (Thought ->)
Chatmoddy, we can passitate to  LLMS:  The water combine this with enternal APIS;  pdf readers, the writers, etc.  in Long Chain to properly make an Application  The make an Application  Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, decides what actor or tool all tomake, feeds the result back into the model until the good is met  That gives us buildry blocks (prompt templates, chains, memories, callbacks, tool wrappers) to brild LLM apple.  TEA (T -> It is a prompting pattern where the model (LLM)
Chatmodoly, we can passitate to  LLMS:  The can combine this with external APIS;  pot readers, the writers, etc  in Long Chown to proporty make an  Application  The can combine / Chown all their together  to make an Application  Agent: In LLM world, an agent  is a loop trad repeatedly reads a  conversation, decides what action  or tool all to make, feeds the result  back into the model until the goal is  met  That gives us buildry blocks  (prompt templates, chains, mamories,  callbacks, tool wrappers) to build LLM apple  TREACT -> It is a prompting  pattern where the model (LLM)  emits an interleaved (Thought ->)

Torsk:	<b>~</b>
information sommerizer.	
-> This is how you would doit way	
Long Cherin madules.	
information = · · · ·	
Some in tornation about en	
Popular person	
Sommary_template=	
given the information: { information	\ <u>{</u>
about a parson, I want you create:	
1. A Short Summary	
2- two intersing fauts	
sommarg - prompt - template - Prompt Templat	ţ
input_vanables = ['intermation]	
templete = summery_templete)	
(m = (hat Open A] (temperateure = 0,	
model_nome = gpt-3.5=tu	۲Ì
Chain = Summary - prompt-template DU	η
V	
Piperym	ЬD —
725 = Chevin. invoke (input = { information	
(retornata);	
prot(res)	

Using Open Source Models with lay chair	<u>^</u>
- Langthoun con be configured to be	
used with Ollama, LLAMA 3,	
Mixtral, etc (local models)	
- However the latency will soffer a	7
the models take forever to run.	
-> pripervinstall langularin-ollama	
(hat Open AI) Object instantiate	
(hat Open AI) Object instant ato	n
uva	
[chatolloma].	
eg: [um = ChatOllema (model= 'llamo	ر کــــــ
Most of the heavy lifting comes from this line	
this bre	
	>
Chain = Summary_prompt_template / Mr	<b>√</b>
StroutputPorner	
here we are une operator overboding	<b>)</b>
hore we are une operator ovalorement of the chain prompt all LLM7	
TO IP	
Strout	zvz Ž
lonks.	1
7 The ice braker application	
-> getters information on	/
agnen individual (tutter,  -> Summarizes it anked 1	n
-> Summarizes it unker!	ニン
-> display to week	
-> display to user	
L> Building this App using:	<i>\\</i>
L> Building thing App using:	<i>\\</i>
Duilding this App using:  1) Chavins	
Daildne thus App use:  1) Chains 2) Agents	
Dailding this App my:  1) Chains 2) Agents 3) Custom Agents	
Daildne thus App use:  1) Chains 2) Agents	

Scrupin 10 Signup
Serne ATI
De-Artice In-A
Scrope Linkod TN Profile and
Scrope Linked In profile and
der
-> Cot data u langan malla
as JSON.
> Pous to LLM if recovery  make decisions/generate  of P based on that profle
mare decisions / general
of parea or the prove
dela.
LLM isn't connected to the Web.
They can be Convidered as a repo
of state information
> Agent/bot can connect to both
internet through tools.
-> tooks are scripts or 3rd pointy
Sernles
-> LLM with a prompt can breakdown
Emospard representations of in
achere those tasks (use tool
Cerlls).
-> These autions can be then thoused
to one another.
-> Agents Think Sime uputh stops
Think ->do taxt.
> Active D
Prompt ->cho formec
D hulpme LANTACHAEN :  dox/2 AGENT
AGENT AGENT
Unes Execute
Unes Exempt
Unes Exemple oxam Idid what
Unes Exempt
Unes Exempt orange (did what were asked me to
Oray Idid what were asked me to  -> Chain of Thought is a prompt
Unes trends  Okayo Idid what  were asked me to

-> Acents we / 1 M + tools to
solve what wer is asking the
agent to solve
The of idea of April is to use Consume mode
The idea of Agril is to use larguage mode to choose a repulse of tenks to
fake.
-> Agents use LLM as a reasoning engine to determe which action
to take.
-> Build a rearch agent.
La Getname from text box
(REACT agest)
> get limed in profle based on UR
We build en rooth too!
Lo Unkedin_lookup_tool.py
-> Import ( prompt template
) tani
from ( reste rest agent
lang chain Agent executor
$\sim$
hub 7  hub 7  hub 2  hub 2  hub 2  hub 2  hub 3
primade