```
# -*- coding: utf-8 -*-
"""Langchain-1

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1XrZtluC5gBuC-g7Mrz7Qtvbxr8lx2ejK
"""

!pip install langchain

!pip install -qU \
    langchain==0.0.292 \
    openai==0.28.0 \
    datasets==2.10.1 \
    pinecone-client==2.2.4 \
    tiktoken==0.5.1

!pip install openai

!pip install cohere

from langchain import PromptTemplate
from langchain import FewShotPromptTemplate

from langchain.llms import OpenAI
from langchain.callbacks import get_openai_callback

import os
os.environ['OPENAI_API_KEY'] = ""

llm = OpenAI(model_name="text-davinci-003", temperature=0)

!pip install deeplake tiktoken

!pip install --upgrade typing-extensions

!pip install -q huggingface_hub

template = """Question: {question}

Answer: """
prompt = PromptTemplate(
    template=template,
    input_variables=['question']
)

# user question
question = "What is the capital city of France?"

from langchain import HuggingFaceHub, LLMChain

hub_llm = HuggingFaceHub(
    repo_id = 'google/flan-t5-large',
    model_kwargs = {'temperature':0},
    huggingfacehub_api_token=''
)

llm_chain = LLMChain(
    prompt = prompt,
    llm = hub_llm
)

print(llm_chain.run(question))

"""ASKING MULTIPLE QUESTIONS"""

qa = [
    {'question':'Who won the last FIFA World Cup'},
    {'question':'Messi or Ronaldo?'}
]

res = llm_chain.generate(qa)
print(res)

multi_template = '''Answer the following questions one at a time.
Question:
{questions}

Answer:
'''

long_prompt = PromptTemplate(template = multi_template, input_variables = ['questions'])

llm_chain = LLMChain(
    prompt=long_prompt,
    llm=hub_llm
)

qs_str = (
    'What is the capital city of Spain?\n'+
    'What is the famous dish in Italy?\n'+
    'Which sport is most famous in India?\n'
)

llm_chain.run(qs_str)

qs_str = (
    "What is the capital city of France?\n" +
    "What is the largest mammal on Earth?\n" +
    "Which gas is most abundant in Earth's atmosphere?\n" +
  "What color is a ripe banana?\n"
)
llm_chain.run(qs_str)

"""TEXT SUMMARIZATION"""
```

```
from langchain.chat_models import ChatOpenAI
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate

llm = ChatOpenAI(
    openai_api_key=os.environ["OPENAI_API_KEY"],
    model='gpt-3.5-turbo'
)

summarization_template = "Summarize the following text: {text}"
summarization_prompt = PromptTemplate(input_variables=['text'], template = summarization_template)
summarization_chain = LLMChain(llm = llm, prompt = summarization_prompt)

text = "For near two decades, Chhetri's fortunes have been synonymous with those of the national team. To the layman, he *is* Indian football. He struts into t

summary = summarization_chain.predict(text=text)

print(summary)

pip install transformers

from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained("gpt2")

print(tokenizer.vocab)

token_ids = tokenizer.encode("This is a sample text to test the tokenizer.")

print(token_ids)

from langchain import OpenAI, PromptTemplate
from langchain.chains.summarize import load_summarize_chain
from langchain.document_loaders import PyPDFLoader

llm = OpenAI(model_name="", temperature=0)

summarize_chain = load_summarize_chain(hub_llm)

!pip install pypdf

document_loader = PyPDFLoader(file_path = '/content/4thaug_inv.pdf')

document = document_loader.load()

summary = summarize_chain(document)

hub_llm = HuggingFaceHub(
    repo_id = 'Falconsai/text_summarization',
    model_kwargs = {'temperature':0},
    huggingfacehub_api_token='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'
)

print(summary['output_text'])

from langchain.chat_models import ChatOpenAI
from langchain.schema import (
    HumanMessage,
    SystemMessage
)

chat = ChatOpenAI(model_name='gpt-4', temperature=0)

messages = [
    SystemMessage(content = "You are a helpful assistant that translates English to Hindi"),
    HumanMessage(content = "Translate the following sentence: I Love Programming")
]

chat(messages)
```