```python
# -*- coding: utf-8 -*-
"""Langchain-2

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1zhmMO88cX3Q7V2a5GcPiEloyiG0Xmpx8

ROLE PROMPTING
"""

!pip install -qU \
    langchain==0.0.208 \
    deeplake==3.6.5 \
    openai==0.27.8 \
    tiktoken==0.4.0 \
    selenium==4.15.2

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

from langchain import PromptTemplate, LLMChain
from langchain.llms import OpenAI

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

template = """
As a futuristic robot band conductor, I need you to help me come up with a song title.
What's a cool song title for a song about {theme} in the year {year}?
"""

prompt = PromptTemplate(
    input_variables=['theme','year'],
    template = template
)

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

input_data = {'theme':'interstellar travel', 'year':'2050'}

response = llm_chain.run(input_data)

print(response)

print("Theme: interstellar travel")
print("Year: 3030")
print("AI-generated song title:", response)

from langchain import FewShotPromptTemplate
from langchain.chat_models import ChatOpenAI

examples = [
    {'color':'red','emotion':'passion'},
    {'color':'blue','emotion':'serenity'},
    {'color':'green','emotion':'tranquility'},

]

example_formatter_template = '''
Color: {color}
Emotion: {emotion}\n
'''

example_prompt = PromptTemplate(
    input_variables = ['color','emotion'],
    template = example_formatter_template
)

few_shot_prompt = FewShotPromptTemplate(
    examples = examples,
    example_prompt = example_prompt,
    prefix="Here are some examples of colors and the emotions associated with them:\n\n",
    suffix="\n\nNow, give a new color, identify the emotion associated with it:\n\nColor:{input}\nEmotion:",
    input_variables=['input'],
    example_separator = '\n',
)

formatted_prompt = few_shot_prompt.format(input='purple')

chain = LLMChain(llm=llm, prompt=PromptTemplate(template=formatted_prompt, input_variables=[]))

response = chain.run({})

print(response)

template_question = '''What is the name of the player who wears number 10 for Real Madrid CF,
Answer:'''
prompt_question = PromptTemplate(template=template_question,input_variables=[])

template_fact = '''Give a brief description of the {player}.
Answer:'''

prompt_fact = PromptTemplate(template=template_fact, input_variables=['player'] )

chain_question = LLMChain(llm=llm, prompt=prompt_question)

response_question = chain_question.run({})

player = response_question.strip()

chain_fact = LLMChain(llm=llm, prompt=prompt_fact)

input_data = {'player':player}
response_fact = chain_fact.run(input_data)

print('Player:', player)
print('Player Description:', response_fact)

examples = [
    {
        "query": "How do you feel today?",
        "answer": "As an AI, I don't have feelings, but I've got jokes!"
    }, {
        "query": "What is the speed of light?",
        "answer": "Fast enough to make a round trip around Earth 7.5 times in one second!"
    }, {
        "query": "What is a quantum computer?",
        "answer": "A magical box that harnesses the power of subatomic particles to solve complex problems."
    }, {
        "query": "Who invented the telephone?",
        "answer": "Alexander Graham Bell, the original 'ringmaster'."
    }, {
        "query": "What programming language is best for AI development?",
        "answer": "Python, because it's the only snake that won't bite."
    }, {
        "query": "What is the capital of France?",
        "answer": "Paris, the city of love and baguettes."
    }, {
        "query": "What is photosynthesis?",
        "answer": "A plant's way of saying 'I'll turn this sunlight into food. You're welcome, Earth.'"
    }, {
        "query": "What is the tallest mountain on Earth?",
        "answer": "Mount Everest, Earth's most impressive bump."
    }, {
        "query": "What is the most abundant element in the universe?",
        "answer": "Hydrogen, the basic building block of cosmic smoothies."
    }, {
        "query": "What is the largest mammal on Earth?",
        "answer": "The blue whale, the original heavyweight champion of the world."
    }, {
        "query": "What is the fastest land animal?",
```

```python
        "answer": "The cheetah, the ultimate sprinter of the animal kingdom."
    }, {
        "query": "What is the square root of 144?",
        "answer": "12, the number of eggs you need for a really big omelette."
    }, {
        "query": "What is the average temperature on Mars?",
        "answer": "Cold enough to make a Martian wish for a sweater and a hot cocoa."
    }
]

example_template = '''
User:{query}
AI: {answer}
'''

example_prompt = PromptTemplate(template=example_template, input_variables=['query','answer'])

prefix = """The following are excerpts from conversations with an AI
assistant. The assistant is typically sarcastic and witty, producing
creative and funny responses to users' questions. Here are some
examples:
"""

suffix = """
User: {query}
AI: """

fshot = FewShotPromptTemplate(
    examples = examples,
    example_prompt = example_prompt,
    prefix=prefix,
    suffix=suffix,
    input_variables = ['query'],
    example_separator = '\n\n'
)

llm = LLMChain(llm=llm, prompt = fshot)

input_data = {"query":"How do I start learning Data Science"}
response = llm.run(input_data)

print(response)

"""USING DEEPLAKE AND SEMANTIC SIMILARITY"""

from langchain.vectorstores import DeepLake
from langchain.embeddings import OpenAIEmbeddings
from langchain.prompts.example_selector import SemanticSimilarityExampleSelector

example_prompt = PromptTemplate(
    input_variables=["input", "output"],
    template="Input: {input}\nOutput: {output}",
)

examples = [
    {"input": "0°C", "output": "32°F"},
    {"input": "10°C", "output": "50°F"},
    {"input": "20°C", "output": "68°F"},
    {"input": "30°C", "output": "86°F"},
    {"input": "40°C", "output": "104°F"},
]

os.environ["ACTIVELOOP_TOKEN"] = ""

my_activeloop_org_id = "ihamzakhan89"
my_activeloop_dataset_name = "langchain_course_fewshot_selector"
dataset_path = f"hub://ihamzakhan89/langchain_course_fewshot_selector"
db = DeepLake(dataset_path=dataset_path)

!pip install deeplake==3.6.5
```