

QUEST 2: In-Context Learning / Prompt Engineering with Large Language Models

⚠ This QUEST requires you to capture a few flags (▶). If you don't know what that is or how to do it, review the instructions in [QUEST 1](#).

Several companies, such as OpenAI, Cohere, and HuggingFace, offer services to prompt Large Language Models. In this QUEST we will be using OpenAI's GPT-3.5 model (text-davinci-003). Before you begin the quest make sure you have an OpenAI account and a key. If you have difficulties, ask Brendan for assistance.

Warmup in Playground

For these exercises, head over to the [OpenAI Playground](#) to solve them. As you solve these exercises, you might be tempted to try too many things in the Playground, because it's super fun! So be careful to not exhaust your introductory \$18 credits before you do rest of this home work. Don't change the temprature or any settings for this homework, unless asked.

- ▶ 1) What is the default temprature? [[capture this flag](#)]
- ▶ 2) Ask GPT-3 to come up with 10 girl baby names for babies from your country of origin. [to [capture this flag](#), submit your prompt in the form]
- ▶ 3) Rewrite your prompt to make the names unique and interesting. [To [capture this flag](#), submit your prompt in the form]

LangChain Exercises

For the rest of the exercises your solutions should be in a [colab notebook](#) and to capture the flags, unless instructed otherwise, you will need to submit the link of the notebook for each flag. They can all be in the same notebook, i.e., no need to create different notebooks, but use the Markdown feature of the notebook to document and explain. You will need to keep your API key ready for this.

You will want to do these steps first before attempting the rest of the questions:

```
✓ [1] !pip install langchain
      !pip install openai
```

```
✓ [2] %env OPENAI_API_KEY= ... # replace ... with your API key
```

Go over the [LangChain](#) documents and figure out how to set temprature for your requests

- ▶ 4) Implement problem 3 using LangChain's LLM and the PromptTemplate classes and check if it works. [[capture this flag](#)]
- ▶ 5) Create a new prompt that takes a baby name and the country of origin, and comes up with a short (made up) biography. [To [capture this flag](#), submit your prompt in the form]

► 6) Follow the [LangChain example to create a custom chain class](#), to create class that returns a list of dicts of {baby_name, biography}. You have to use the prompts you created in 4 and 5. [\[capture this flag\]](#)

ArXiv Bulletin

Every day several hundred, if not more, papers appear on ArXiv. For this task you will implement an ArXiv bulletin that gives a list of paper titles and why I should read it in **one sentence**. Your final output should look something like this (the exact papers will be different):

- **Summarizing Encyclopedic Term Descriptions on the Web**
Why to read: This paper presents a summarization method to produce a single text from multiple descriptions to concisely describe a term from different viewpoints.
- **Unsupervised Topic Adaptation for Lecture Speech Retrieval**
Why to read: This paper presents a novel approach to improve the quality of a cross-media information retrieval system by adapting acoustic and language models for automatic speech recognition.
- ...

Download a sample of the NLP ArXiv dataset from [here](#). It has metadata for 100 NLP papers as JSON records. For this exercise, you will **randomly pick 10 records** to show a proof of concept. Here's a starter code for you that you can copy paste in a notebook.

```
import json
import random
import gdown

gdown.download('https://drive.google.com/uc?id=1_oPRNSW7QWd1Us-APMV5Y7h6RxU_8gF')
with open('cs.cl.sample100.json') as f:
    data = f.readlines()

parsed = [json.loads(x) for x in data]
sample10 = random.choices(parsed, k=10)
```

You will have to use LangChain for this problem and come up with a suitable zero-shot prompt for "why". Be creative with your prompts and use the prompting best practices as your guide. You can experiment with a few prompt variations on for a single paper in Playground before you use it in your code. Make sure your code runs at T=0.

► [\[Capture the flag\]](#) for this by entering the Colab notebook URL for your work.

Congratulations! You are done with this homework.