Quick tip: Review the prerequisites before you run the lab

Start Lab 01:00:00

Orchestrating LLMs with LangChain: Challenge Lab

Lab 1 hour No cost Intermediate

Rate Lab

This lab may incorporate AI tools to support your learning.

Challenge Lab Overview

This lab will challenge you to perform actions and automation across products. Instead of following step-by-step instructions, you are given a common business scenario and a

tasks correctly.

When you take a Challenge Lab, you will not be taught Google Cloud concepts. You will need to use your skills to assess how to build the solution to the challenge presented. This lab is only recommended for students who have those skills. Are you up for the challenge?

Previous Next

Objective

Vertex AI Gemini foundational models — Text, Chat, and Embeddings — are officially integrated with the LangChain Python SDK, making it convenient to build applications on top of Vertex AI Gemini models. You can now create Generative AI applications by combining the power of Vertex AI models with the ease of use and flexibility of LangChain.

LangChain framework for developing Language Model (LLM) applications. Through this challenge, participants will showcase their ability to utilize pre-trained models effectively.

Setup and requirements

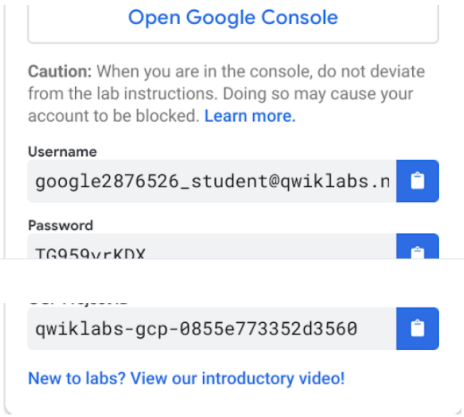
For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 02:00:00 and make sure you can finish in that time block.

beginning.

3. When ready, click START LAB .
4. Note your lab credentials. You will use them to sign in to the Google Cloud

Console.



5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

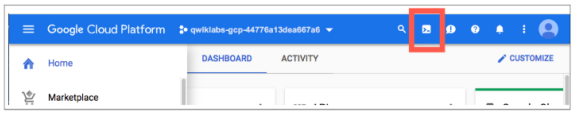
7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

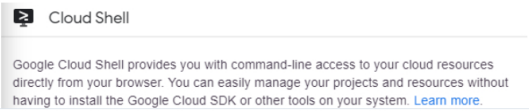
Activate Cloud Shell

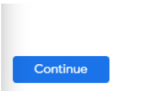
Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.





It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

```
Credentialed accounts:
 - <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the gcloud command-line tool overview.

Challenge Scenario

Here is a company overview as provided on the Cymbal Shops's website.

> Cymbal Shops is an American retail chain headquartered in Minneapolis that sells homeware, electronics, and clothing.
>
> Founded in 1974, Cymbal Shops started out as Cymbal Air, selling AC systems manufactured by Cymbal Group in Minnesota and neighboring states. The company quickly expanded into domestic merchandise in order to satisfy the need for quality products at an affordable price in the midst of a recession.
>
> Cymbal Shops' broad product assortment was once a benefit - capturing planned purchases and last minute splurges. In recent years, however, the company has struggled to adapt to the acceleration in e-commerce. Digitally native companies are on the rise and Cymbal Shops must implement significant changes in order to keep pace and maintain relevance. Today, Cymbal Shops operates 714 stores across North America and reported $15 billion in revenue in 2019. They currently employ 80,400 employees across the United States and Canada.

> It is inspired by clients like: Bed Bath & Beyond, Best Buy, Home Depot, Nordstrom.

Your challenge

Cymbal Shops is likely looking to leverage Generative AI to automate or enhance the process of creating catalog content, including product descriptions, reviews, and specifications. This could involve using Large Language Models (LLMs) to generate creative and informative text summaries, thereby streamlining the catalog creation process and potentially improving the quality and uniqueness of the content.

As a software engineer, you are tasked with creating a Proof of Concept (POC) using LangChain, a framework for developing LLM applications, to demonstrate the capability of generating concise summaries with a pre-trained model. Your goal is to showcase the potential of this technology in enhancing product descriptions, customer reviews, and technical specifications for upcoming catalogs. The challenge involves loading the model, preparing relevant data, and designing a prompt template to efficiently generate text summarizations, all through the application of LangChain. This effort paves the way

Task 1. Open the notebook in Vertex AI Workbench

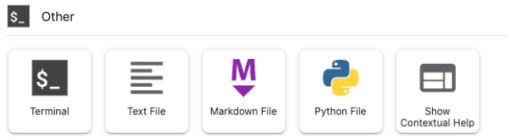
1. In the Google Cloud Console, on the **Navigation menu**, click **Vertex AI > Workbench**.

2. Find the `Workbench instance name` instance and click on the **Open JupyterLab** button.

The JupyterLab interface for your Workbench instance will open in a new browser tab.

Task 2. Set up the notebook and install packages

1. Once your JupyterLab interface opens select **Terminal**.



2. Use `gsutil` command to copy the Jupyter Notebook file from `item fill at the lab start` into the home directory of your Vertex AI Workbench instance's JupyterLab interface and open it.

3. In the **Select Kernel** dialog, choose **Python 3** from the list of available kernels.

Click **Check my progress** to verify the objective.

Download Jupyter notebook file

Check my progress

Install packages and import necessary libraries

In the challenge notebook, run all the cells for **Task 2** to install and import necessary packages and libraries.

Task 3. Load the model

The Gemini foundation models are optimized for a variety of natural language tasks, including sentiment analysis, entity extraction, and content creation. The types of content that the Gemini models can generate include document summaries, answers to questions, and labels that classify content.

For this challenge, you should load the pre-trained text generation model called `gemini-1.5-pro`.

Therefore, in the next cell under **Task 2** of the notebook, write an equivalent command to load the pre-trained `Gemini` model. In later tasks, we will use this model to create a shorter version of a document that incorporates pertinent information from the original text.

Gemini model is fine-tuned to follow natural language instructions and is suitable for a variety of language tasks, such as classification, summarization, and extraction.

Click here for hint!

Preparing data files

In this sub-task, you should preprocess the data files required for summarization to make it suitable for input to the text generation model.

Therefore, in this task you should;

1. In the next cell under, write a code snippet that downloads a PDF file from the following URL:

```
https://services.google.com/fh/files/misc/practitioners_guide
```

> The PDF file contains a document that takes a deeper dive into the themes of scale and automation to illustrate the requirements for building and operationalizing ML systems.

Click here for hint!

3. Finally, load the PDF file and split it into individual pages.

Click here for hint!

Click **Check my progress** to verify the objective.

Prepare data files

Check my progress

Task 4. Text Summarization

In this task, let's use the pre-trained model and data files loaded in earlier tasks to generate summaries of large text. This lab uses the method of `Stuffing`, which is the simplest method to pass data to a language model. It "stuffs" text into the prompt as context in a way that all of the relevant information can be processed by the model to get what you want.

Therefore in this task you should;

1. Utilize the following template to create a prompt for summarizing a given text.

```
prompt_template = """Write a concise summary of the
following text delimited by triple backquotes.
              Return your response in bullet points which
covers the key points of the text.
              ```{text}```
              BULLET POINT SUMMARY:
  """

prompt = PromptTemplate(template=prompt_template,
```

2. Set up a summarization chain using the `stuff` method. Incorporate the model loaded earlier into the summarization chain to enhance the quality of the summary.

3. Take the first three pages of a document as the input text and apply the summarization chain with the initialized model to generate a concise summary.

Click here for hint!

Click **Check my progress** to verify the objective.

Text Summarization

Check my progress
```

Task 5. Tool calls

In this task, let's use the same model and add tools to perform certain tasks. Specifically, we will add a multiplication function as a tool, so that when the user asks a multiplication question, instead of the model trying to solve it through chain of thought, it actually tells you to call the function.

1. Create the multiplication function and annotate it with a tool. The `multiply` function takes in two integers `item fill at the lab start` and `item fill at the lab start` and performs a multiplication `a*b` returning the result from `langchain_core.tools import tool`.

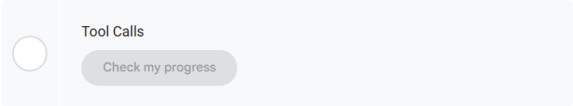
2. Add the tools to the LLM created earlier and invoke it with the following query. Print the result in the console.

```
from langchain_core.messages import HumanMessage,
ToolMessage, AIMessage
query = "What is 3 * 12?"
```

3. Iterate through the tools in the response, invoke the tools and append the response to the messages object.

4. Invoke the LLM with the solution of the tool and the original message and print the final user response.

Click **Check my progress** to verify the objective.



Congratulations!

Congratulations! You've successfully completed the challenge lab. Your adept skills in crafting effective prompts, loading pre-trained models, and generating concise summaries showcase a high level of proficiency in utilizing LLMs for text summarization within the LangChain framework.

Manual Last Updated November 20, 2024

Lab Last Tested November 20, 2024

Copyright 2023 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.