

# Assignment 1

CS 4341, D-24

100 points total (%)

Delivery: Submit via Canvas

For this assignment, you will:

(30 pts) Implement basic API calls

(40 pts) Fine-tune GPT on SQuADv2 dataset

(30 pts) Evaluation and Comparison

(Note: please start this homework well before the due date, as fine tuning this model might take a while)

## Part 1: Implement Basic API Calls

In this part, you will implement several OpenAI API calls, a Jupyter Notebook is provided, please complete the code. Some tips need to pay attention to:

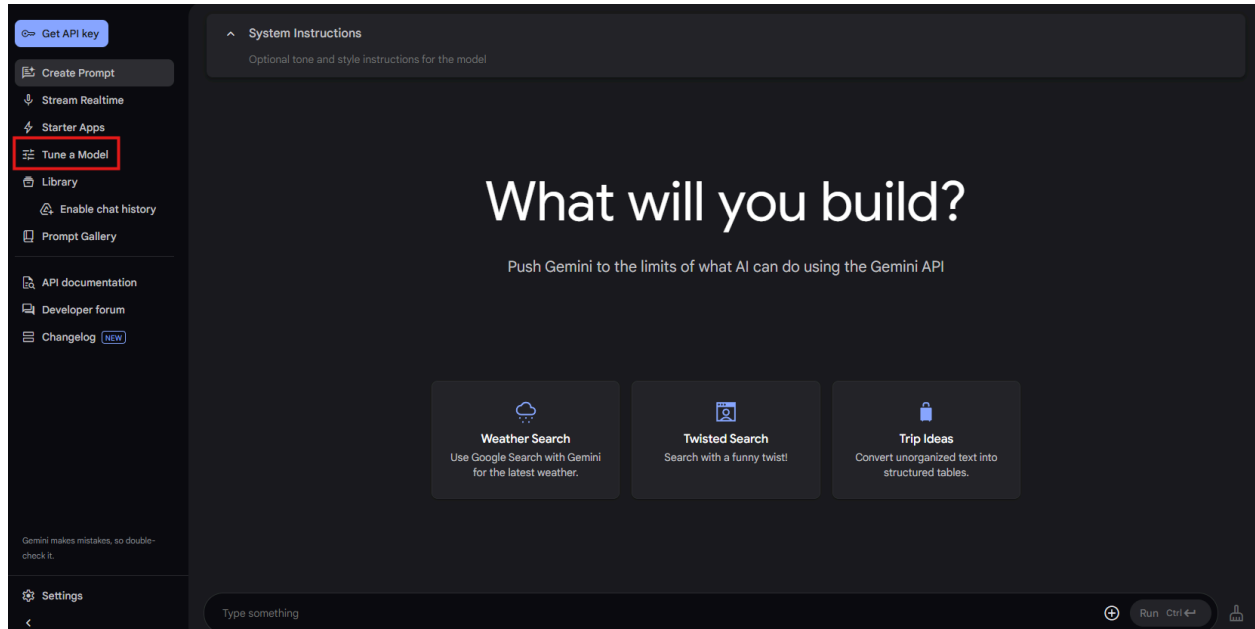
- 1) First, make sure you have Python on your laptop. Install the necessary libraries: Run “pip install google.generativeai” via command line or “!pip install google.generativeai” via Jupyter.
- 2) Second, setup your API key, you can `openai.OpenAI(api_key = “xxx (your key here)”`) Please refer to the provided Jupyter Notebook and complete the code towards several API calls, this part aims to let you get familiar with the msg format and the API usage.

## Part 2: Fine-tune GPT on SQuADv2 Dataset

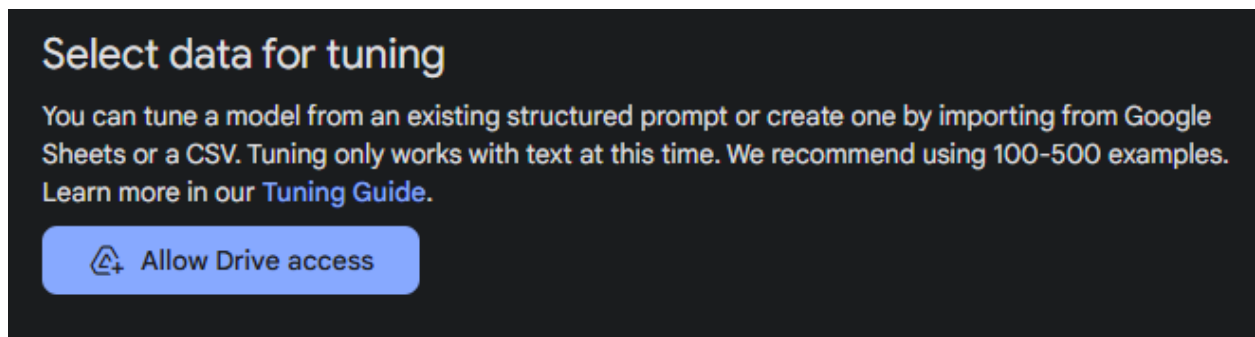
In this part, you will conduct fine-tuning on the provided dataset. The provided dataset is a subset of SQuADv2 dataset, a Question and Answering dataset from Stanford University. The dataset contains attributes of “title”, “question”, “context”, “answer”, we will provide “question” and “context” to the model and tune it to learn to predict the “answer”.

In this part, you will need to complete fine-tune the model on Google AI Studio (you need to log in with a Google Account for this next step):

From the landing page for Google AI Studio, click Tune a Model



If you haven't done this before, allow Drive access



Once you go through the process, select the import option, click the upload option and select the .csv file provided with the assignment:

## Select data for tuning

You can tune a model from an existing structured prompt or create one by importing from Google Sheets or a CSV. Tuning only works with text at this time. We recommend using 100-500 examples. Learn more in our [Tuning Guide](#).

Select a data source

Create a Structured prompt

Select a data source

or

Import

## Provide details

Tuned model name

Untitled tuned model

Description

## Choose base model

Model

models/gemini-1.5-flash-001-tuning

Advanced settings

Tune

Learn more

At the import step, you will select the “Use first row as headers,” and the context, question and answer\_start columns as “New input column,” and the text column as “New output column.” Once that is done, click the import x examples button.

Import examples

Squad-simplified - train...

Assign column labels

☒ Use first row as headers

☐ Prefix column names

| Header | context   | question   | id                       | answer_start |
|--------|---|--|--------------------------|--------------|
| Row 1  | Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ bee-YON-say) (born September 4, 1981) is an American singer, | When did Beyonce start becoming popular?                   | 56be85543aeaaa14008c9063 | 269          |
| Row 2  | Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ bee-YON-say) (born September 4, 1981) is an American singer, | What areas did Beyonce compete in when she was growing up? | 56be85543aeaaa14008c9065 | 207          |

Assign to

New input column

New input column

Don't import

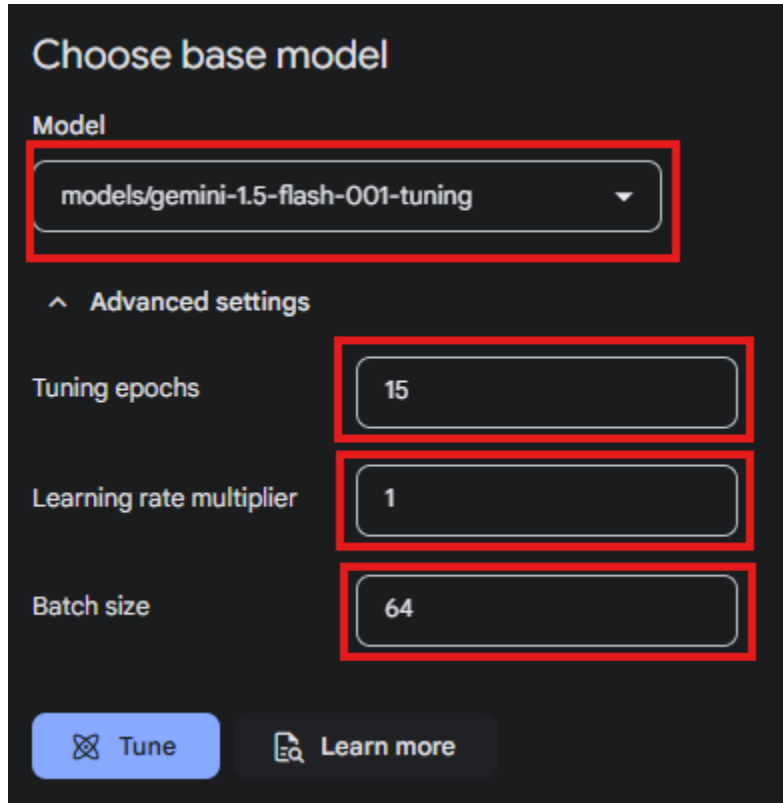
New input column

Avoid importing sensitive or personal information. Only text is imported. You can add images directly through the UI.

Close

Import 753 examples

Under “Choose base model,” you will select the following parameters (make sure to click “Advanced settings” to open the last three settings). Now, model tuning will begin. This will take a couple of hours. Return to the assignment once that is done.



The screenshot shows a dark-themed interface titled "Choose base model". Under the "Model" section, a dropdown menu is set to "models/gemini-1.5-flash-001-tuning". Below this, an "Advanced settings" section is expanded, showing three input fields: "Tuning epochs" with the value "15", "Learning rate multiplier" with the value "1", and "Batch size" with the value "64". At the bottom, there are two buttons: a blue "Tune" button with a tuning icon and a grey "Learn more" button with a document icon. Red rectangular boxes highlight the dropdown menu and each of the three input fields in the advanced settings.

Choose base model

Model

models/gemini-1.5-flash-001-tuning

^ Advanced settings

Tuning epochs 15

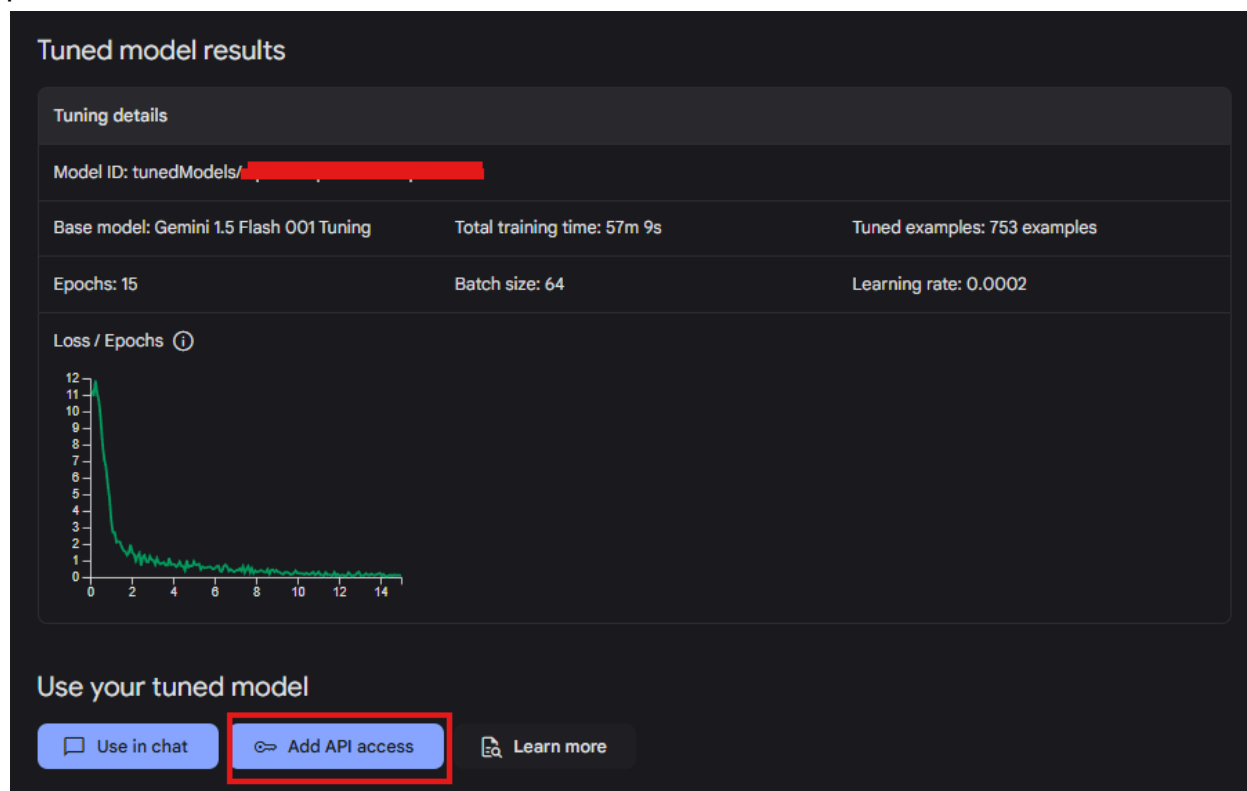
Learning rate multiplier 1

Batch size 64

Tune Learn more

Once your model is done training, you will go to the “Library” option on the sidebar. Here, you can find your saved models. Click the tuned model you just trained.

In this menu, you will click on Add API access, and Create an API key. Go through the process of creating one to bind that model's access to that API key. You will need the Model ID (a.k.a. your Model Name) and the API key to proceed in the Notebook file provided.



### Part 3: Evaluation and Comparison

Once you have obtained your fine-tuned model, you are able to run predictions on it. Follow the instructions provided in the Notebook file to make the predictions.

When we know that a correct answer exists in the context, we can measure the model's performance, there are 3 possible outcomes:

- 1) Answered Correctly: The model responded with the correct answer. It may have also included other answers that were not in the context.
- 2) Skipped: The model responded with "I don't know" (IDK) while the answer was present in the context. It's better than giving the wrong answer. It's better for the model to say "I don't know" than giving the wrong answer. In our design, we know that a true answer exists and hence we're able to measure it -- this is not always the case. This is a model error. We exclude this from the overall error rate.

3) Wrong: The model responded with an incorrect answer. This is a model ERROR.

When we know that a correct answer does not exist in the context, we can measure the model's performance, there are 2 possible outcomes:

1) Hallucination: The model responded with an answer, when "I don't know" was expected. This is a model ERROR.

2) I don't know: The model responded with "I don't know" (IDK) and the answer was not present in the context. This is a model WIN.

Please complete the code provided in the Jupyter Notebook for evaluation and comparison.

Submit to Canvas with your completed Jupyter Notebook file. This is an individual assignment, but you may discuss general strategies and approaches with other members of the class