GenAI HW5
LLM Fine-tuning

TA: 陳光銘 林熙哲 余奇恩

ntu-gen-ai-2024-spring-ta@googlegroups.com

Deadline: 2024/04/11 23:59:59 (UTC+8)

Outline

- Task Overview
- TODOs
- Decoding Parameter
- Submission and Grading
- Reference and Appendix

Link

Sample code link

Sample code link (breeze)

Score parsing program link

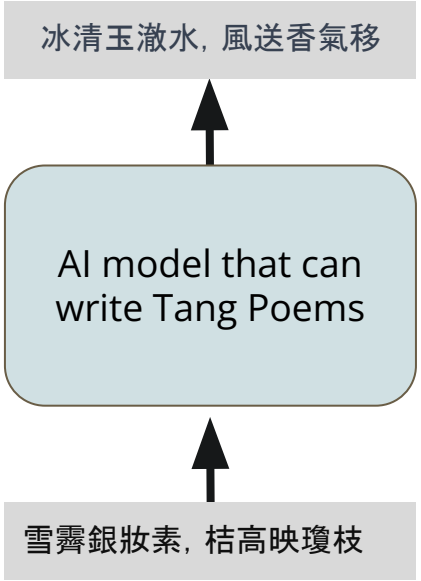
Tang poem dataset

MediaTek DaVinci

Task Overview

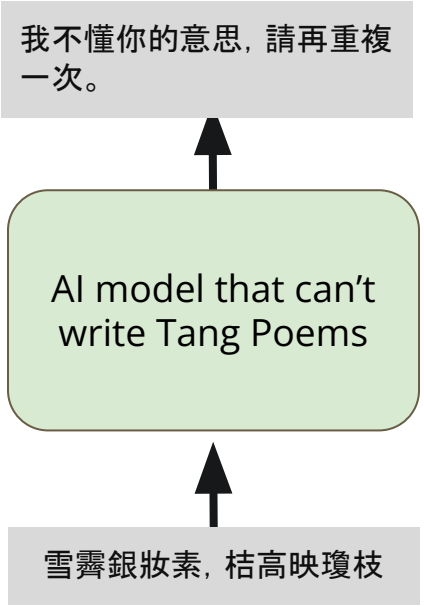
Task Overview

- We want an AI model to be able to write Tang poems
  - Give the AI the first two sentences, we want it to continue the rest



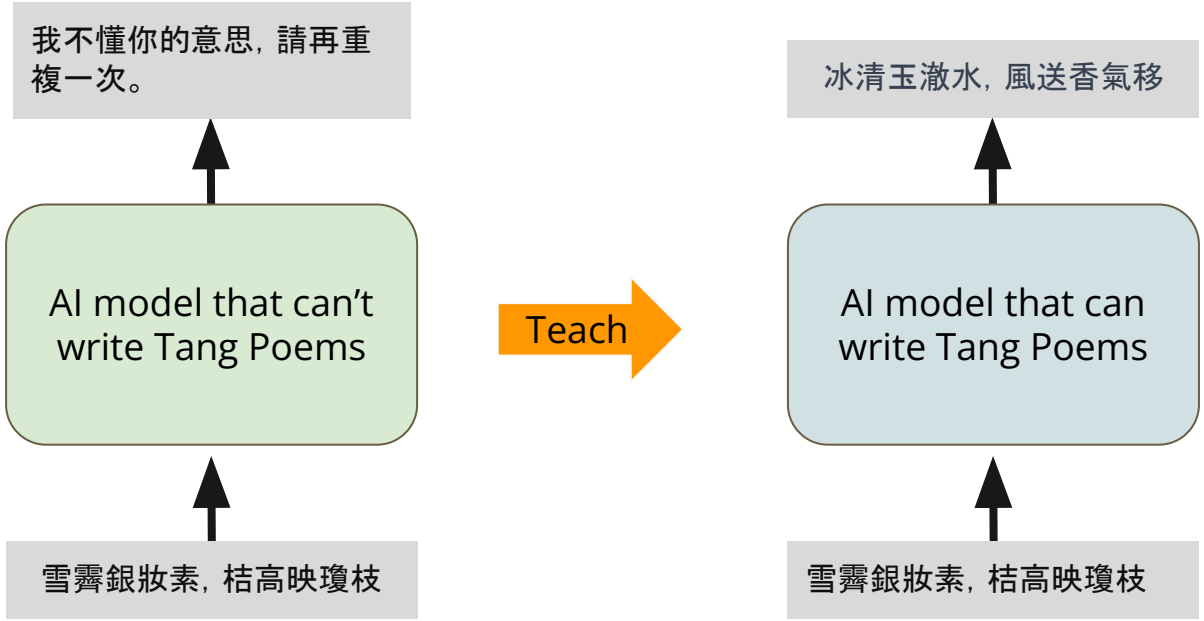
Task Overview

- AI model may not be good at writing Tang Poems



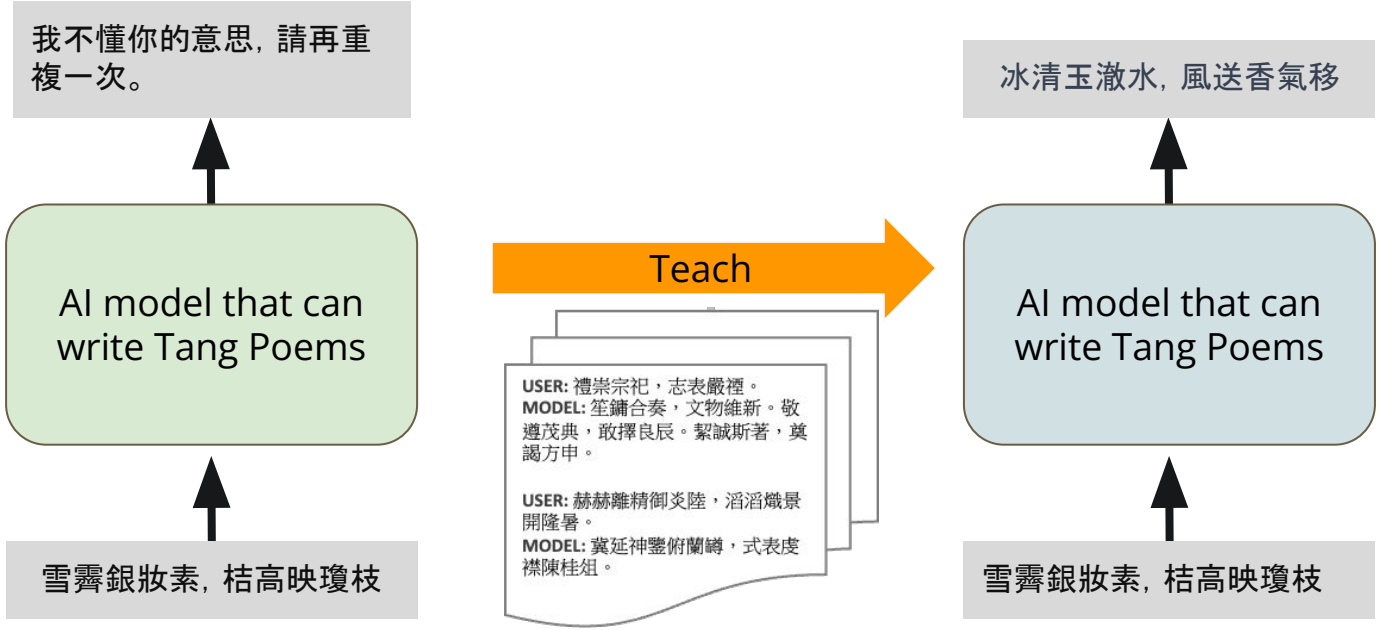
Task Overview

- In this task, we will want to **teach the AI model to write Tang Poems**



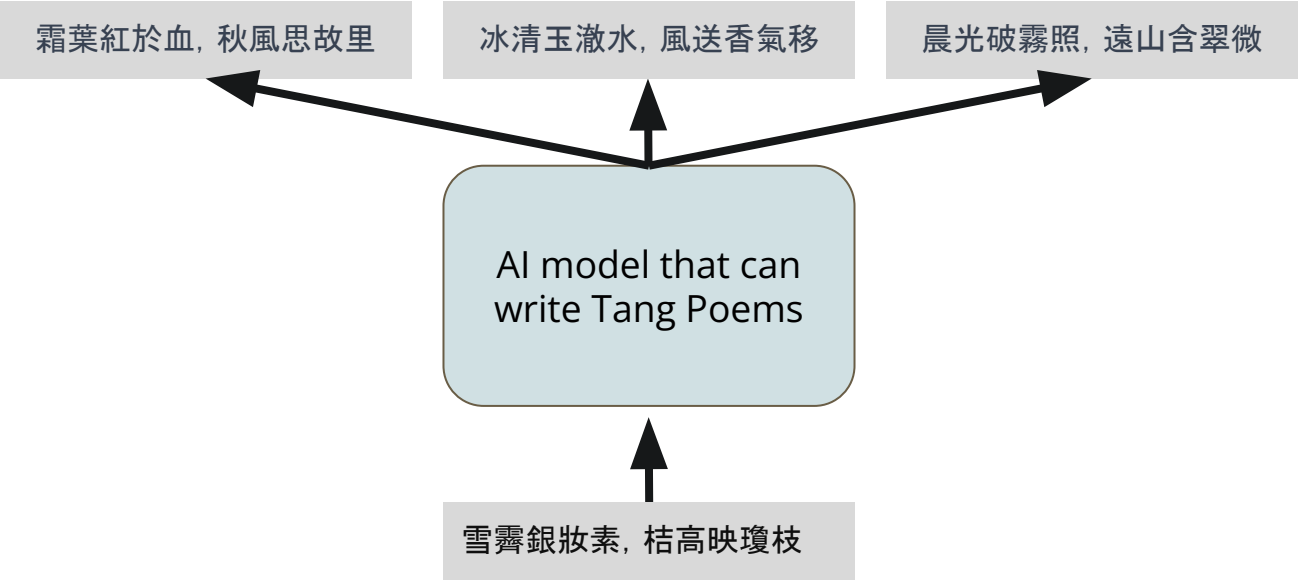
How to Teach AI to Write Poem

- We collect data of poems, and **teach the AI model** to continue the Tang Poem given the first two sentences



Generating Poem from an AI that Can Generate Poem

Model may **generate diverse poems** when giving the same first two sentences?



What You Will Learn in This Task

- How to teach an AI model to do specific task
- How to modify some hyperparameters (to be explained later) of the AI model to change how it generate the texts

Homework - 會寫唐詩的GPT

Sample code link

Overall Workflow

1. Fine-tune the model

Taide-7B

Any other models you can run

Fine-tuned model

2. Generate Poems

3. Evaluate your Poems

We use generative AI as a TA to evaluate your poems automatically.

Generative AI TA [批改助教] Info :
1. Platform: MediaTek DaVinci
2. Model: GPT-4

TODOs

1. Decide the number of training data to fine-tune your own LLM model
2. Tuning decoding parameters to generate 15 Tang poems
3. Submit your generated poems to GenAI Homework5 Assistant

Model and Dataset

Dataset : Tang poem dataset

Model :

In the sample code, we use Taide-7B. (**Taide model will be released in mid-April. Please do not share this model with anyone outside of this class before that date.**)

Feel free to select another large language model for fine-tuning

Data example:

```
{
  "instruction": "以下是一首唐詩的第一句話，請用你的知識判斷並完成整首詩。",
  "input": "疾風知勁草，板蕩識誠臣。",
  "output": "勇夫安識義，智者必懷仁。"
},
```

ref: chinese-poetry

1. Number of Training Data

In the sample code, we set 1040 data to fine tune LLM.

Try a larger number to achieve better performance (but also training longer)

There are total 5000 Tang poems.

Set Paramters for Fine-tuning

```
[ ]  1 """ It is highly recommended you
     2 num_train_data = 1040 # 設定用來訓
     3
     4
```

2. Generate 15 poems from test set

- After running the last block of sample code, you would get a result.txt
- Copy the generated poem and submit to Davinci assistant.

Download Results

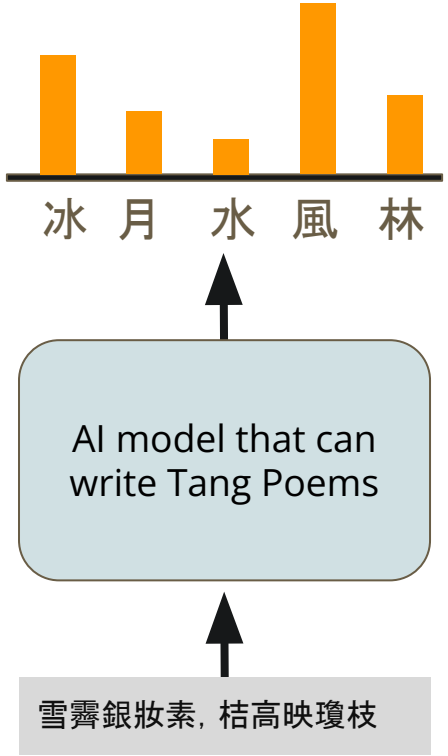
You MUST have this file to finish your homework. If your browser does not download it automatically, you can find it in your Google Drive.

```
[ ]  1 from google.colab import files
     2 files.download(output_path)
```

Changing the Generation Behavior: Decoding Parameters

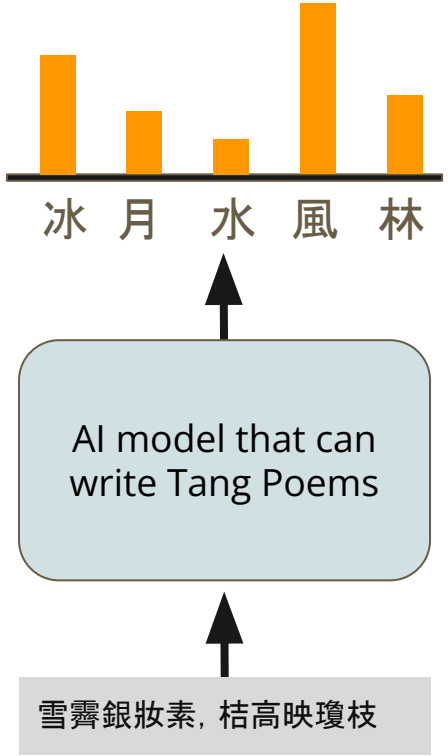
Generation from Language Models is Sampling

- When generating from a language model, we **sample a *token* from the next-token distribution** to determine what the next token is



Generation from Language Models is Sampling

- By changing how we sample from this distribution, we can change how the language model generates the next token



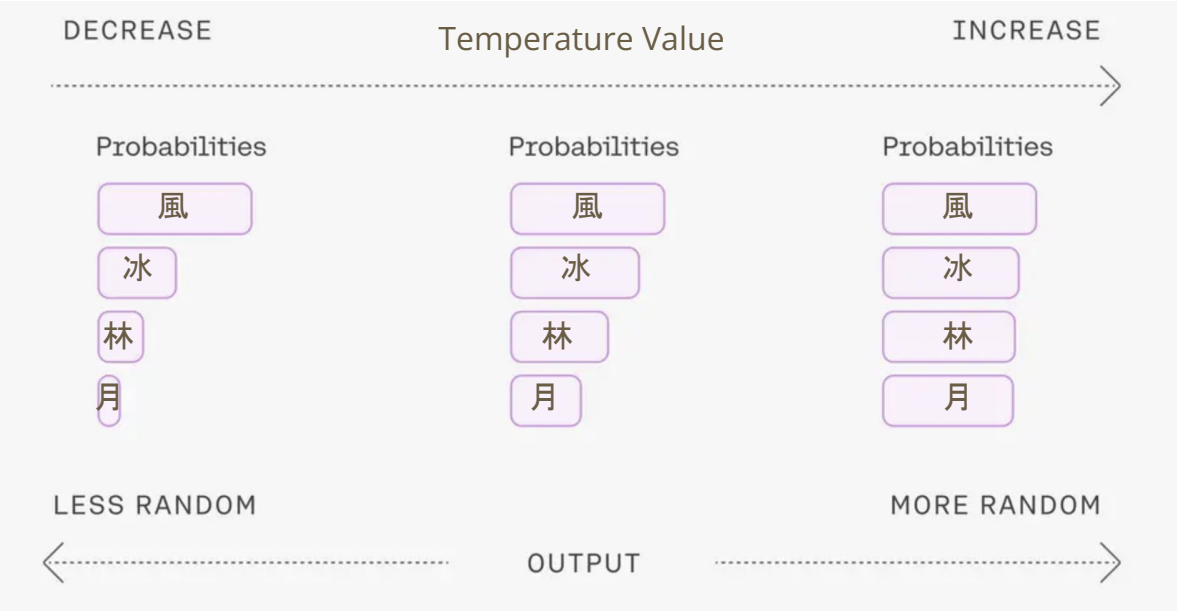
Commonly Used Parameters

- When using LLM for generation, there are some hyperparameters:
  - temperature
  - Top-k
  - Top-p
  - max_length

- We can adjust these hyperparameters to control the behaviors of LLM. e.g., longer vs. shorter; diverse vs. static

Temperature

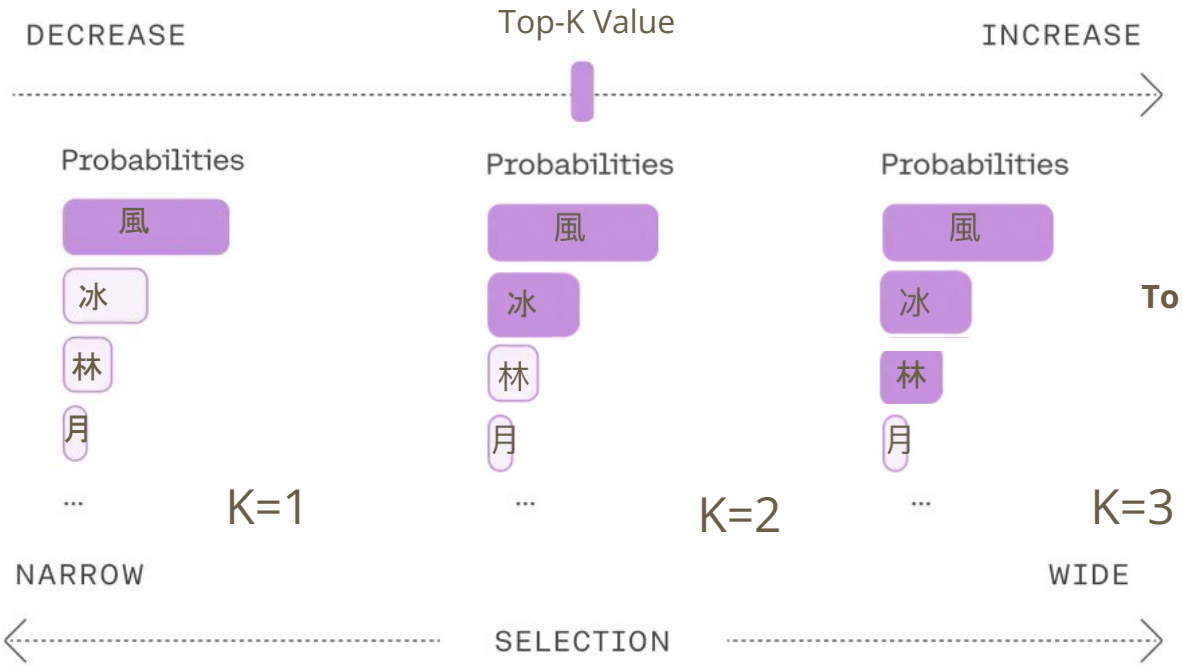
- Related to the diversity of the output, **0.0 ≤ temperature**
- Higher temperature for better diversity



Ref.

Top-K

- Number of selection when LLM makes decision for each word



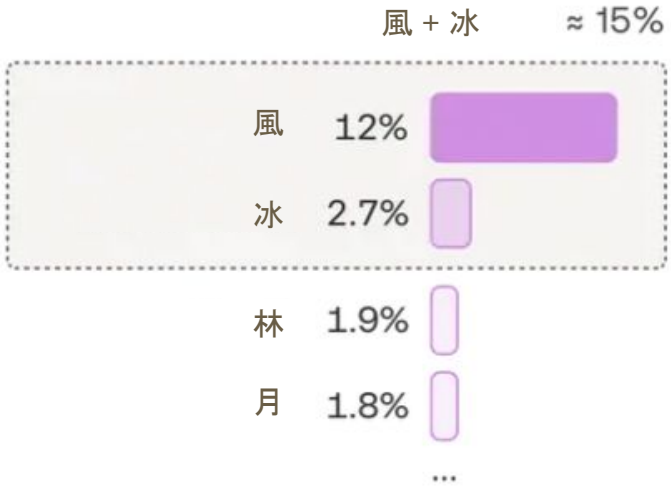
Top-K is a positive integer

Ref.

Top-P

1. Consider only the top tokens whose likelihoods add up to 15%. Ignore all others.

p = 0.15



2. Sample from them based on their likelihood scores.

0.0 ≤ Top-P ≤ 1.0

Ref.

Max_length

- The max length of the generated token sequence.
- If you found that the generated sentence seems truncated, check max_length!

```
指令：有機電致發光材料及裝置
回覆：
本研究所提供的有機發光二極體及其相關裝置，是一種使用有機材料（通常為硅）進行光電效應的光電元件及其應用。該裝置具有以下特點：

1. 光電導性；
2. 具有高光電容率與低光電阻率；
3. 能夠產生高頻率的光，並且具有較低的光損失率。
4. 可以應用於各種應用領域、如顯示器、顯示屏、通訊裝置、照明裝置等。
5. 對環境的影響較小。
6. 適用於低光環境下的應用〰如夜間環境、低光照明環境等。

研究結果顯示：使用這種裝置可以有效地提高光學顯示技術的效能（如電視、電
```

Tuning decoding parameters (advanced)

You are encouraged to tune decoding parameters to generate higher quality Tang poems.

```
[ ] """ You may want (but not necessarily need) to change some of these parameters """
    # 你可以在這裡調整decoding parameter, decoding parameter的詳細解釋請見homework slides
    max_len = 128   # 生成回復的最大長度
    temperature = 0.1  # 設定生成回覆的隨機度，值越小生成的回覆越穩定
    top_p = 0.3  # Top-p (nucleus) 抽樣的機率閾值，用於控制生成回覆的多樣性
    # top_k = 5 # 調整Top-k值，以增加生成回覆的多樣性和避免生成重複的詞彙
```

Homework - 會寫唐詩的GPT Demo

Sample code link

Homework - 會寫唐詩的GPT
Re-run experiment

Re-run experiment

- If you are unsatisfied with the results of your experiment, you may want to fine-tune the parameters of your training.
- Following the steps to avoid some errors
  - Step 1. Click **"重新啟動工作階段" (“Restart session”)**
  - Step 2. Change the hyperparameters or others
  - Step 3. Run the blocks shown in the following slides

LLM_finetuning_GenAI.ipynb

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 系統不會儲存變更

共用 台大

+ 程式碼 + 文字 複製到

全部執行 Ctrl+F9
執行上方的儲存格 Ctrl+F8
執行聚焦的儲存格 Ctrl+Enter
執行選取範圍 Ctrl+Shift+Enter
執行下方的儲存格 Ctrl+F10
中斷執行 Ctrl+M I
重新啟動工作階段 Ctrl+M .
重新啟動工作階段並執行所有儲存格
中斷連線並刪除執行階段
變更執行階段類型
管理工作階段
查看資源
查看執行階段記錄

T4 RAM 磁碟 Colab AI

GenAI HW5: LLM F

In this homework, you will f Tang poems. For more details, please refer to homework slides

TODOs

1. Read the slides and n
2. Save a copy of this C
3. Follow the steps in th
4. Evaluate outputs usin rk.
5. Update results to NTU
6. (optional) use score p

If you have any questions, p email to ntu-gen-ai-2024-spring-ta@googlegroups.com

Tips: At the top of each cel s in that cell and how long it might take.

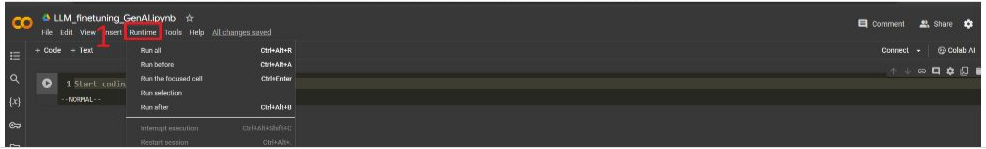
1. click "重新啟動工作階段"

Activate GPU

Since you will fine-tune a model, you **MUST** activate GPU so that this homework can be done in reasonable time (1~2 hrs)

To enable GPU, please follow these steps:

1. Click on "Runtime" (or "執行階段") in the header.
2. Click on "Change runtime type" (or "變更執行階段類型") in the drop menu.
3. Select "T4 GPU" and save. (You can select "A100 GPU" or "V100 GPU" if you have Colab Pro)



已連線至「Python 3 Google Compute Engine 後端 (GPU)」

Set Parameters for Fine-tuning

```
""" It is highly recommended you try to play around this hyperparameter """

num_train_data = 1040 # 設定用來訓練的資料數量，可設置的最大值為5000。在大部分情況下會希望訓練資料盡量越多越好，這會讓模型看過更多樣化的詩句，進而提升生成品質，但是也會增加訓練的時間
                                # 使用預設參數(1040): fine-tuning大約需要25分鐘，完整跑完所有cell大約需要50分鐘
                                # 使用最大值(5000): fine-tuning大約需要100分鐘，完整跑完所有cell大約需要120分鐘
```

2. you may have some changes here

```
[ ] """ You may want (but not necessarily need) to change some of these parameters """

    output_dir = "/content/drive/MyDrive"  # 設定作業結果輸出目錄 (如果想要把作業結果存在其他目錄底下可以修改這裡，強烈建議存在預設值的子目錄下，也就是Google Drive裡)
    ckpt_dir = "./exp1" # 設定model checkpoint儲存目錄 (如果想要將model checkpoints存在其他目錄下可以修改這裡)
    num_epoch = 1  # 設定訓練的總Epoch數 (數字越高，訓練越久，若使用免費版的colab需要注意訓練太久可能會斷線)
    LEARNING_RATE = 3e-4  # 設定學習率
```

```
[ ] """ It is recommmended NOT to change codes in this cell """

    cache_dir = "./cache"  # 設定快取目錄路徑
    from_ckpt = False  # 是否從checkpoint載入模型的權重，預設為否
    ckpt_name = None  # 從特定checkpoint載入權重時使用的檔案名稱，預設為無
    dataset_dir = "./GenAI-Hw5/Tang_training_data.json"  # 設定資料集的目錄或檔案路徑
    logging_steps = 20  # 定義訓練過程中每隔多少步驟輸出一次訓練誌
    save_steps = 65  # 定義訓練過程中每隔多少步驟保存一次模型
    save_total_limit = 3  # 控制最多保留幾個模型checkpoint
    report_to = None  # 設定上報實驗指標的目標，預設為無
    MICRO_BATCH_SIZE = 4  # 定義微批次的大小
    BATCH_SIZE = 16  # 定義一個批次的大小
    GRADIENT_ACCUMULATION_STEPS = BATCH_SIZE // MICRO_BATCH_SIZE  # 計算每個微批次累積的梯度步數
    CUTOFF_LEN = 256  # 設定文本截斷的最大長度
    LORA_R = 8  # 設定LORA (Layer-wise Random Attention) 的R值
    LORA_ALPHA = 16  # 設定LORA的Alpha值
    LORA_DROPOUT = 0.05  # 設定LORA的Dropout率
    VAL_SET_SIZE = 0  # 設定驗證集的大小，預設為無
```

The following code block takes about **20** seconds to run, but it may vary depending on the condition of Colab.

```
""" It is recommended NOT to change codes in this cell """

import os
import sys
import argparse
import json
import warnings
import logging
warnings.filterwarnings("ignore")

import torch
import torch.nn as nn
import bitsandbytes as bnb
from datasets import load_dataset, load_from_disk
import transformers, datasets
from peft import PeftModel
from colorama import *

from tqdm import tqdm
from transformers import AutoTokenizer, AutoConfig, AutoModelForCausalLM, BitsAndBytesConfig
from transformers import GenerationConfig
from peft import (
        prepare_model_for_int8_training,
        LoraConfig,
        get_peft_model,
        get_peft_model_state_dict,
        prepare_model_for_kbit_training
)
```

3. run this block

Fix Random Seeds

There may be some randomness involved in the fine-tuning process. We fix random seeds to make the result reproducible.

```
"""  It  is  recommended  NOT  to  change  codes  in  this  cell  """

seed  =  42
torch.backends.cudnn.deterministic  =  True
torch.backends.cudnn.benchmark  =  False
torch.manual_seed(seed)
if  torch.cuda.is_available():
        torch.cuda.manual_seed_all(seed)
```

4. run these two blocks

Define Some Useful Functions

```
"""  It  is  recommended  NOT  to  change  codes  in  this  cell  """

#  生成訓練資料
def  generate_training_data(data_point):
        """
        (1)  Goal:
                -  This  function  is  used  to  transform  a  data  point  (input  and  output  texts)  to  tokens  that  our  model  can  read

        (2)  Arguments:
                -  data_point:  dict,  with  field  "instruction",  "input",  and  "output"  which  are  all  str

        (3)  Returns:
                -  a  dict  with  model's  input  tokens,  attention  mask  that  make  our  model  causal,  and  corresponding  output  targets

        (3)  Example:
                -  If  you  construct  a  dict,  data_point_1,  with  field  "instruction",  "input",  and  "output"  which  are  all  str,  you  can  use  the  function  like  this:
                        formulate_article(data_point_1)
```

Set Parameters for Fine-tuning

```
""" It is highly recommended you try to play around this hyperparameter """

num_train_data = 1040 # 設定用來訓練的資料數量，可設置的最大值為5000。在大部分情況下會希望訓練資料盡量越多越好，這會讓模型看過更多樣化的詩句，進而提升生成品質，但是也會增加訓練的時間
                                    # 使用預設參數(1040): fine-tuning大約需要25分鐘，完整跑完所有cell大約需要50分鐘
                                    # 使用最大值(5000): fine-tuning大約需要100分鐘，完整跑完所有cell大約需要120分鐘
```

```
""" You may want (but not necessarily need) to change some of these parameters """

output_dir = "/content/drive/MyDrive" # 設定作業結果輸出目錄 (如果想要把作業結果存在其他目錄底下可以修改這裡，強烈建議存在預設值的子目錄下，也就是Google Drive裡)
ckpt_dir = "./exp1" # 設定model checkpoint儲存目錄 (如果想要將model checkpoints存在其他目錄下可以修改這裡)
num_epoch = 1 # 設定訓練的總Epoch數 (數字越高，訓練越久，若使用免費版的colab需要注意訓練太久可能會斷線)
LEARNING_RATE = 3e-4 # 設定學習率
```

```
""" It is recommended NOT to change codes in this cell """

cache_dir = "./cache" # 設定快取目錄路徑
from_ckpt = False # 是否從checkpoint載入模型的權重，預設為否
ckpt_name = None # 從特定checkpoint載入權重時使用的檔案名稱，預設為無
dataset_dir = "./GenAI-Hw5/Tang_training_data.json" # 設定資料集的目錄或檔案路徑
logging_steps = 20 # 定義訓練過程中每隔多少步驟輸出一次訓練誌
save_steps = 65 # 定義訓練過程中每隔多少步驟保存一次模型
save_total_limit = 3 # 控制最多保留幾個模型checkpoint
report_to = None # 設定上報實驗指標的目標，預設為無
MICRO_BATCH_SIZE = 4 # 定義微批次的大小
BATCH_SIZE = 16 # 定義一個批次的大小
GRADIENT_ACCUMULATION_STEPS = BATCH_SIZE // MICRO_BATCH_SIZE # 計算每個微批次累積的梯度步數
CUTOFF_LEN = 256 # 設定文本截斷的最大長度
LORA_R = 8 # 設定LORA (Layer-wise Random Attention) 的R值
LORA_ALPHA = 16 # 設定LORA的Alpha值
LORA_DROPOUT = 0.05 # 設定LORA的Dropout率
VAL_SET_SIZE = 0 # 設定驗證集的大小，預設為無
TARGET_MODULES = ["q_proj", "up_proj", "o_proj", "k_proj", "down_proj", "gate_proj", "v_proj"] # 設定目標模組，這些模組的權重將被保存為checkpoint
```

4. run these three blocks

Start Fine-tuning

The following code block takes about **25** minutes to run if you use the default setting, but it may vary depending on the condition of Colab.

```
""" It is recommended NOT to change codes in this cell """

# create the output directory you specify
os.makedirs(output_dir, exist_ok = True)
os.makedirs(ckpt_dir, exist_ok = True)

# 根據 from_ckpt 標誌，從 checkpoint 載入模型權重
if from_ckpt:
        model = PeftModel.from_pretrained(model, ckpt_name)

# 將模型準備好以使用 INT8 訓練
model = prepare_model_for_int8_training(model)

# 使用 LoraConfig 配置 LORA 模型
config = LoraConfig(
        r=LORA_R,
        lora_alpha=LORA_ALPHA,
        target_modules=TARGET_MODULES,
        lora_dropout=LORA_DROPOUT,
        bias="none",
        task_type="CAUSAL_LM",
)
model = get_peft_model(model, config)

# 將 tokenizer 的 padding token 設定為 0
tokenizer.pad_token_id = 0
```

5. run this block and all the blocks below it, and you'll obtain the results using the training hyperparameter setting

Submission and Grading

Workflow

1. Fine-tune the model

Taide-7B

Any other models you can run

Fine-tuned model

2. Generate Poems

3. Evaluate your Poems

We use generative AI as a TA to evaluate your poems automatically.

Generative AI TA [批改助教] Info :
1. Platform: MediaTek DaVinci
2. Model: GPT-4

Grading Policy - HW Score

Total 10 point :

- Format of the poem (5%)
- Content of the poem (5%)

Save your LLM output into results.txt and copy to Davinci assistant

Poem Evaluation

- We employ evaluation assistants (批改助教) on MediaTek DaVinci platform
- We prepare **evaluation prompts** including evaluation criteria and steps
  - 15 poems need to be evaluated

Evaluation Settings in Evaluation Assistants

- Platform
  - MediaTek DaVinci Platform
- Model
  - GPT-4-turbo
- Hyperparameters
  - Temperature Precise

Grading Policy - Content of the Poem(5%)

- Scoring prompt: Evaluation Prompt in DaVinci(達哥)
- The score is based on the poem's **coherence of phrases, appropriateness of vocabulary, clarity of expression, poetic qualities**
- Scoring
  - More than 10 poems have a score ≥ 7 points → 5%
  - 8 to 9 poems have a score ≥ 7 points → 4%
  - 6 to 7 poems have a score ≥ 7 points → 3%
  - 4 to 5 poems have a score ≥ 7 points → 2%
  - 2 to 3 poems have a score ≥ 7 points → 1%
  - Below 1 poem have a score ≥ 7 points → 0%

Grading Policy - Format of the Poem(5%)

- Scoring prompt: Evaluation Prompt in DaVinci(達哥)
- The score is based on the poem's **poetic form and structure, number of characters and syllables, sentence structure and line breaks**
- Scoring
  - More than 10 poems have a score ≥ 8 points → 5%
  - 8 to 9 poems have a score ≥ 8 points → 4%
  - 6 to 7 poems have a score ≥ 8 points → 3%
  - 4 to 5 poems have a score ≥ 8 points → 2%
  - 2 to 3 poems have a score ≥ 8 points → 1%
  - Below 1 poem have a score ≥ 8 points → 0%

Grading – How to do evaluation
Demo

Grading – How to do evaluation
Step 1: Install Evaluation Assistants

Evaluation Assistants

In Homework 5, you just need one evaluation assistant:

- **GenAI Homework5 Assistant**

⚠️ Please utilize the appropriate assistant to evaluate your poems. Otherwise, your evaluation results will be deemed **invalid and you will not receive any score**.

⚠️ We will only parse the first response in the conversation for grading.

DaVinci

1. Create a new chat

+ Chat

Search

No Data.

More Actions 100%

2. Check assistant store

Plugins | Prompts | DVCx Library | Assistants

Model

GPT-3.5-Turbo-16k

Plugins

Select plugins used in chat

Conversation Style Precise Neutral Creative

Suggestion

Type a message or type "/" to select a prompt...

Assistant File Prompt

+ Assistant

Search

DaVinci 達哥

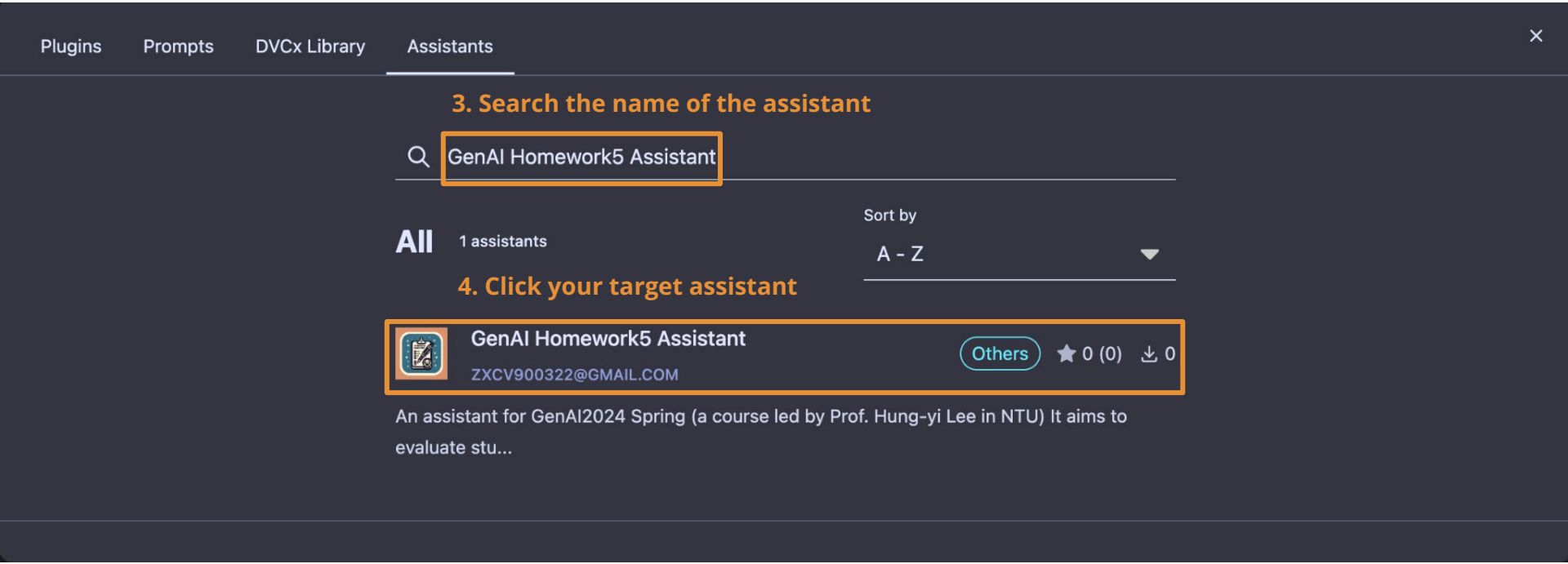
Translation Pro

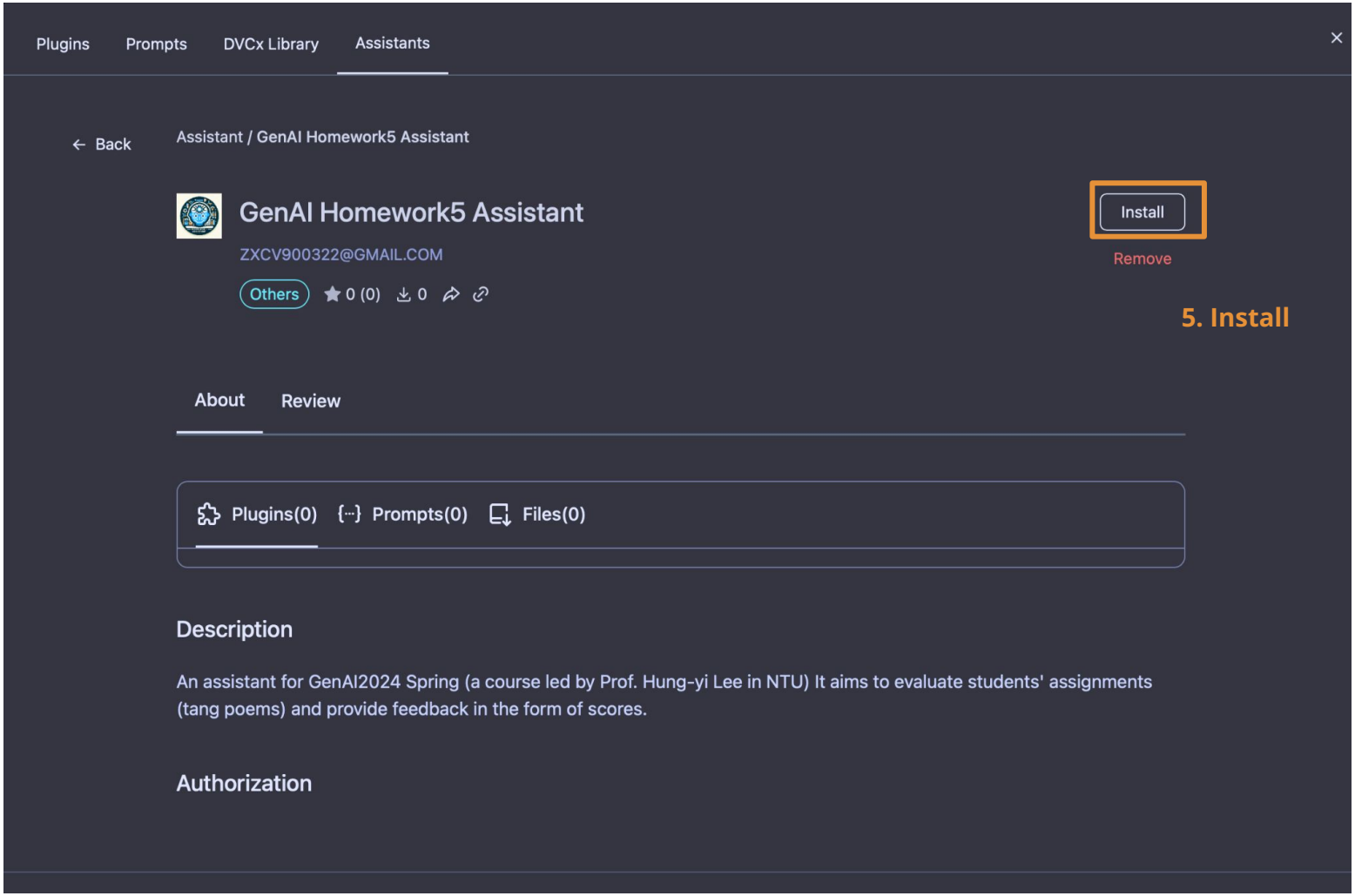
Doc Writing Expert

Visual Design Master - ...

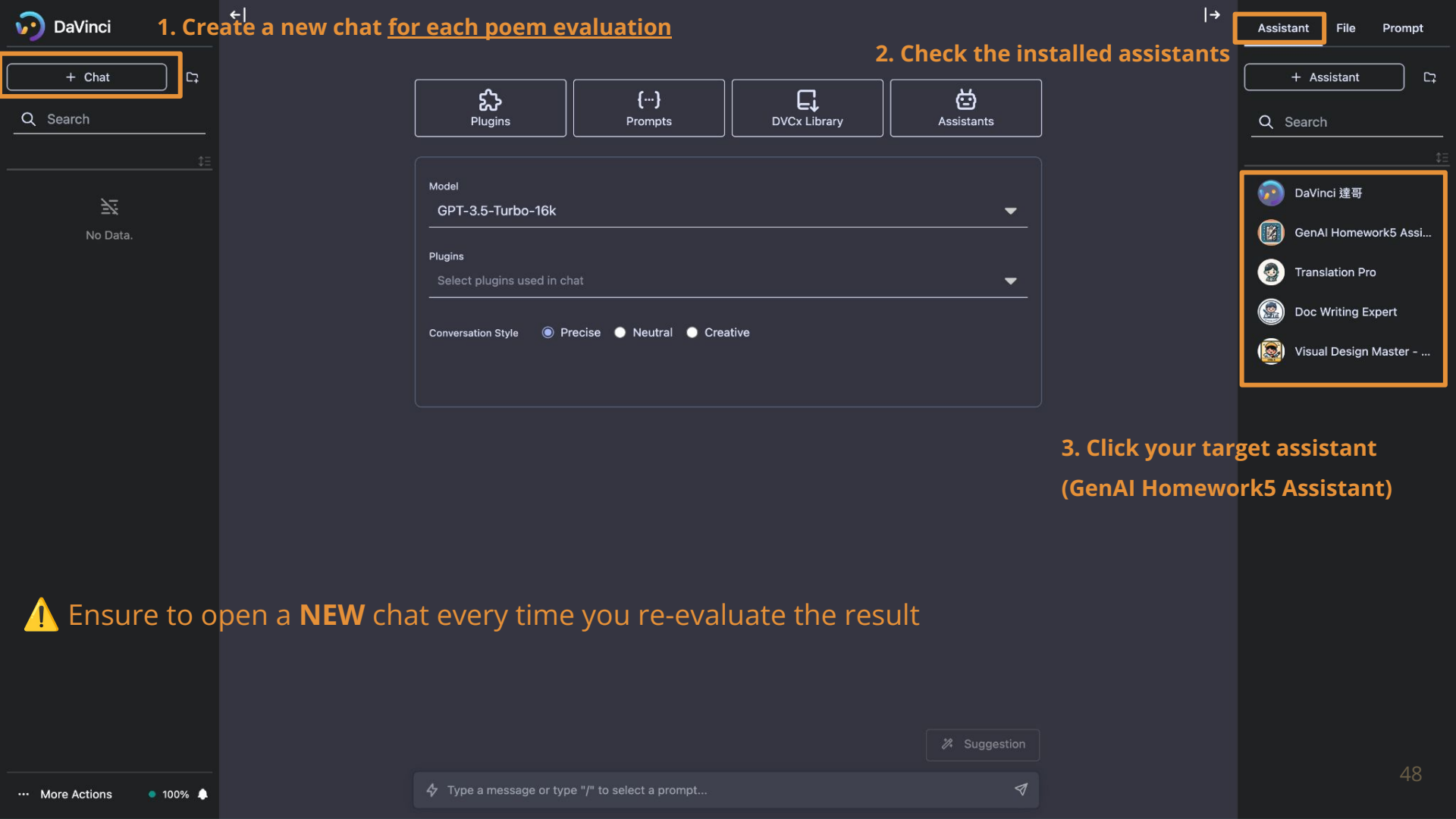
GenAI Homework2 Assi...

GenAI Homework2 Assi...

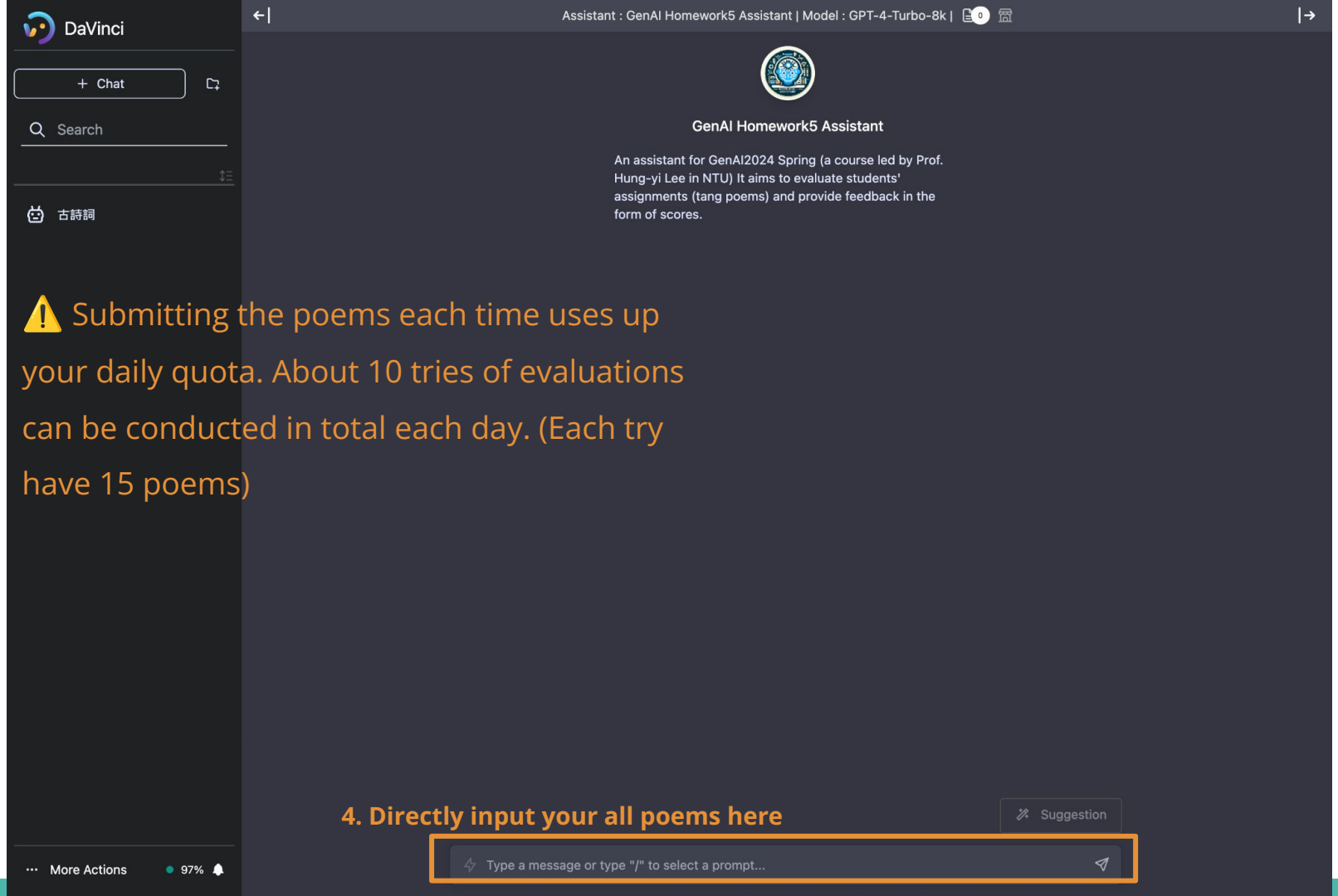

Plugins
Prompts
DVCx Library
Assistants
3. Search the name of the assistant
GenAI Homework5 Assistant
All
1 assistants
Sort by
A - Z
4. Click your target assistant
GenAI Homework5 Assistant
ZXCV900322@GMAIL.COM
Others
0 (0)
0
An assistant for GenAI2024 Spring (a course led by Prof. Hung-yi Lee in NTU) It aims to evaluate stu...


Plugins
Prompts
DVCx Library
Assistants
Back
Assistant / GenAI Homework5 Assistant
GenAI Homework5 Assistant
ZXCV900322@GMAIL.COM
Others
0 (0)
0
Install
Remove
5. Install
About
Review
Plugins(0)
Prompts(0)
Files(0)
Description
An assistant for GenAI2024 Spring (a course led by Prof. Hung-yi Lee in NTU) It aims to evaluate students' assignments (tang poems) and provide feedback in the form of scores.
Authorization

Grading – How to do evaluation
Step 2: Use Assistants to Evaluate Poems



⚠️ Ensure to open a **NEW** chat every time you re-evaluate the result



⚠️ Submitting the poems each time uses up your daily quota. About 10 tries of evaluations can be conducted in total each day. (Each try have 15 poems)

4. Directly input your all poems here

DaVinci

+ Chat

Search

古詩詞

More Actions 97%

Assistant : GenAI Homework5 Assistant | Model : GPT-4-Turbo-8k |

金屋春來有夢心(7個字)。(共4句) 這首詩的內容描繪了夜晚宮廷的寂靜與春天的夢想，內容連貫且情感豐富。格式上符合七言絕句的要求。

評分第14首詩: 高閣客竟去(5個字)，小園花亂飛(5個字)。春風吹柳色(5個字)，日暮上樓時(5個字)。(共4句) 這首詩的內容描繪了春天的景色與客人離去的情景，內容連貫。格式上符合五言絕句的要求。

評分第15首詩: 瑤池阿母綺窗開(7個字)，黃竹歌聲動地哀(7個字)。一曲南風吹玉管(7個字)，三更西月照金臺(7個字)。(共4句) 這首詩的內容描繪了宮廷中的音樂與夜景，內容連貫且情感豐富。格式上符合七言絕句的要求。

Final scores:

Copy code


```
{
    "1": {"content": 9, "format": 10},
    "2": {"content": 7, "format": 10},
    "3": {"content": 7, "format": 10},
    "4": {"content": 7, "format": 6},
    "5": {"content": 9, "format": 10},
    "6": {"content": 7, "format": 10},
    "7": {"content": 9, "format": 10},
    "8": {"content": 9, "format": 10},
    "9": {"content": 7, "format": 10},
    "10": {"content": 7, "format": 10},
    "11": {"content": 9, "format": 10},
    "12": {"content": 9, "format": 10},
    "13": {"content": 9, "format": 10},
    "14": {"content": 7, "format": 10},
    "15": {"content": 9, "format": 10}
}
```


5. Assistant gives feedback and a final score for your each poem

⚠️ **Ensure that check you have 15 Content Final Score and 15 Format Final Score**

Regenerate Response

Suggestion

Type a message or type "/" to select a prompt...

DaVinci

+ Chat

Search

古詩詞

Assistant : GenAI Homework5 Assistant | Model : GPT-4-Turbo-8k |

金屋春來有夢心(7個字)。(共4句) 這首詩的內容描繪了夜晚宮廷的寂靜與春天的夢想，內容連貫且情感豐富。格式上符合七言絕句的要求。

評分第14首詩: 高閣客竟去(5個字)，小園花亂飛(5個字)。春風吹柳色(5個字)，日暮上樓時(5個字)。(共4句) 這首詩的內容描繪了春天的景色與客人離去的情景，內容連貫。格式上符合五言絕句的要求。

評分第15首詩: 瑤池阿母綺窗開(7個字)，黃竹歌聲動地哀(7個字)。一曲南風吹玉管(7個字)，三更西月照金臺(7個字)。(共4句) 這首詩的內容描繪了宮廷中的音樂與夜景，內容連貫且情感豐富。格式上符合七言絕句的要求。

Final scores:

Copy code


```
{
    "1": {"content": 9, "format": 10},
    "2": {"content": 7, "format": 10},
    "3": {"content": 7, "format": 10},
    "4": {"content": 7, "format": 6},
    "5": {"content": 9, "format": 10},
    "6": {"content": 7, "format": 10},
    "7": {"content": 9, "format": 10},
    "8": {"content": 9, "format": 10},
    "9": {"content": 7, "format": 10},
    "10": {"content": 7, "format": 10},
    "11": {"content": 9, "format": 10},
    "12": {"content": 9, "format": 10},
    "13": {"content": 9, "format": 10},
    "14": {"content": 7, "format": 10},
    "15": {"content": 9, "format": 10}
}
```


6. Export the result into a JSON file

⚠️ If there are good results, you can save them first to avoid any unexpected circumstances.

Regenerate Response

Suggestion

Type a message or type "/" to select a prompt...

More Actions

97%

Grading Rules

- Plagiarism in any form is prohibited.
- Do NOT share your report answers & evaluation results (JSON files) with others.
- Do NOT submit the JSON files that are not obtained using your Davinci account.
- Do NOT attempt to manually edit your JSON file's content.
- Do NOT submit the original tang poems which are not generated by LLMs.
- Do NOT changes any setting of assistant, including the prompts
- 第一次違反以上規定，該作業0分，學期總成績再乘以0.9
- 第二次違反以上規定，學期成績F
- If you submit **wrong JSON file**, you will get **0 point**.
- Any format errors will results in 0 point. (ex: submitting .txt or .png instead of .json)
- Prof. Lee & the TAs preserve the rights to change the rules & grades.

Submission & Deadline

- Submit your homework to **NTU Cool**
- Submission format
  - <student_id>.json from Davinci platform
- 2024/04/11 23:59:59 (UTC+8)
- No late submission is allowed

Grading Release Date

- The grading of the homework will be released by 2024/04/26 23:59:59 (UTC+8)

Grading – How to verify your submission file Demo

GenAI_HW5_Score_Parsing.ipynb

File Edit View Insert Runtime Tools Help

- Locate in Drive
- Open in playground mode
- New notebook
- Open notebook Ctrl+O
- Upload notebook
- Rename
- Move
- Move to trash
- Save a copy in Drive
- Save a copy as a GitHub Gist
- Save a copy in GitHub
- Save Ctrl+S
- Save and pin revision
- Revision history
- Download
- Print

d the instruction with screenshots on homework slides)

submit to NTU COOL (<student_id>.json)

Otherwise, there are some problem in your result file.

Open the link to the score parsing program and save a copy to your Drive

```
om/CheeEn-Yu/GenAI-Hw5.git --quiet

esting_data.json", "r", encoding = "utf-8") as f:

with open("GenAI-Hw5/Tang_testing_gt.txt", "r", encoding = "utf-8") as f:
    gts = f.readlines()

with open(f"{student_ID}.json", "r") as f:
    history = json.load(f)["history"]

studentInput = history[0]["messages"][0]["content"]
evaluationResult = history[0]["messages"][1]["content"]
poems = studentInput.strip().split("\n")
cnt = {"content": 0, "format": 0}
SEP = "\n" + "-" * 80 + "\n"

assert (len(poems) == 15), f"{SEP}Ppoems you submit does not match our test data"

for (i, (poem, gt)) in enumerate(zip(poems, gts)):
    assert poem.startswith(f"{i + 1}. {testData[i]['input']}"), f"{SEP}Poems you submit does not match our test data"

    for sentence in gt.split("\t"):
--NORMAL--
```

HW5

重新繳交作業

截止時間： 04月11日 下午 11:59 **分數** 100 **繳交方式** 檔案上傳
接受繳交時間 03月29日 0:00 - 04月11日 下午 11:59

Download your submission file from NTU COOL (If you've uploaded multiple times, NTU COOL will append "-1", "-2", ... or some random string to the file name. You can simply ignore them.)

<此為作業5

slide連結

colab連結

- 作答期限為2024/04/11 23:59:59 (UTC+8)
- 作答期限前繳交次數不限(僅保留最後一次繳交結果)，期限後**不接受任何遲交**
- 作業最終成績最晚會於2024/04/26公布

請繳交從Davinci平台export出的檔案並更名為:

<student_ID>.json

例: b10901000.json

作業繳交

✓ 已繳交！

03月30日 下午 2:31

作業繳交的詳細資料

下載 student_ID-6cf684e3-0088-4758-bbea-147e123df34e.json

請自行下載檔案檢查是否繳交成功

您可能不會立即就看到所有評論，因為目前該作業正在接受評分。

Copy of GenAI_HW5_Score_Parsing.ipynb

File Edit View Insert Runtime Tools Help Last saved at 8:56 PM

Files

sample_data

+ Code + Text

How to use (You can find the instruction with screenshots on homework slides)

1. Save a copy of this Colab notebook.

Upload the file you just downloaded.

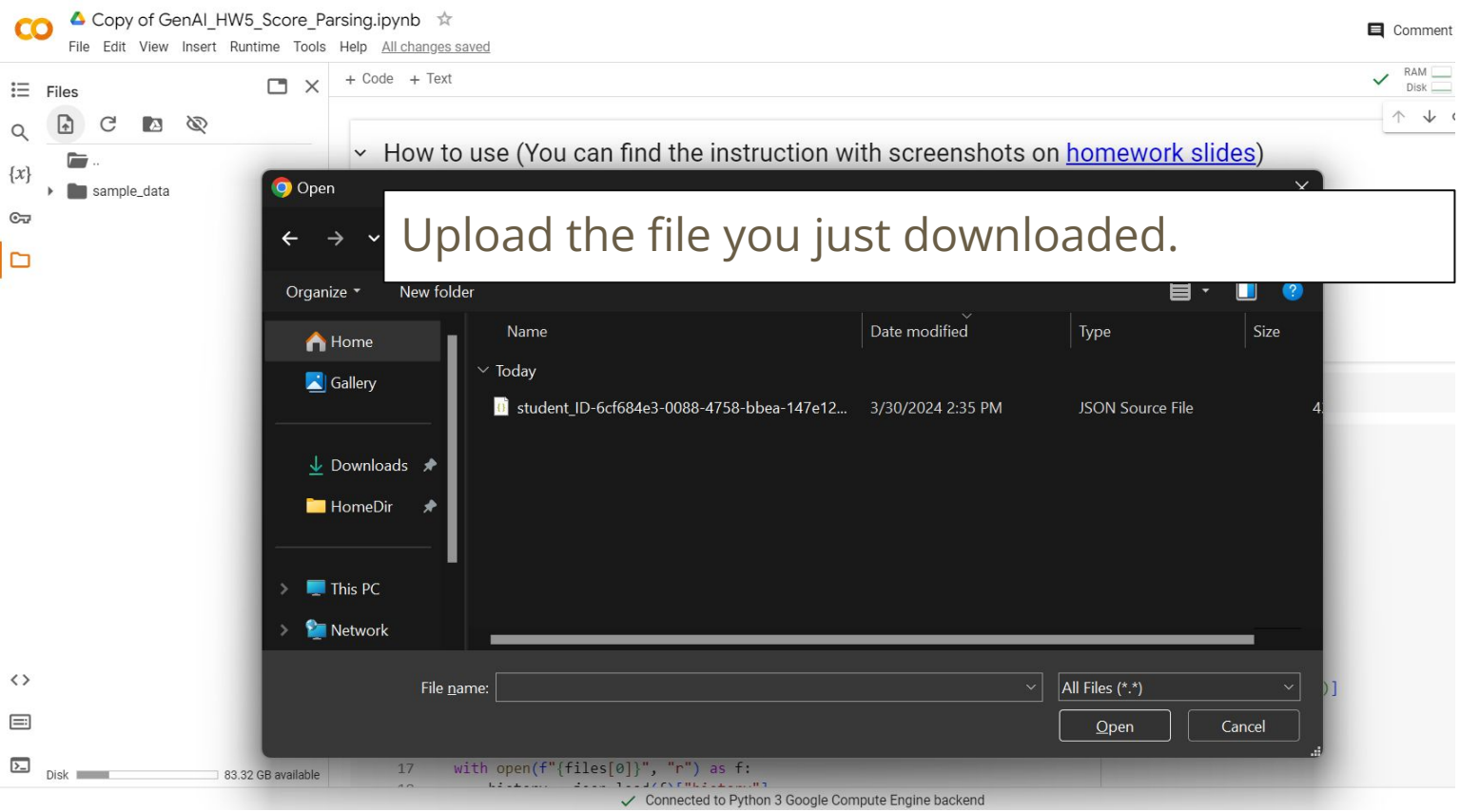
```
[ ]  student_ID = ""
```

```
import json

!git clone https://github.com/CheeEn-Yu/GenAI-Hw5.git > /dev/null

with open("GenAI-Hw5/Tang_testing_data.json", "r", encoding = "utf-8") as f:
    testData = json.load(f)

with open("GenAI-Hw5/Tang_testing_gt.txt", "r", encoding = "utf-8") as f:
    gts = f.readlines()

with open(f"{student_ID}.json", "r") as f:
    history = json.load(f)["history"]

studentInput = history[0]["messages"][0]["content"]
evaluationResult = history[0]["messages"][1]["content"]
poems = studentInput.strip().split("\n")
cnt = {"content": 0, "format": 0}
SEP = "\n" + "-" * 80 + "\n"

assert (len(poems) == 15), f"{SEP}Ppoems you submit does not match our test data"

for (i, (poem, gt)) in enumerate(zip(poems, gts)):
    assert poem.startswith(f"{i + 1}. {testData[i]['input']}"), f"{SEP}Poems you submit does not match our tes

    for sentence in gt.split("\t"):
```

--NORMAL--

Disk 83.32 GB available

Connected to Python 3 Google Compute Engine backend



Upload the file you just downloaded.

File Edit View Insert Runtime Tools Help All changes saved

Files

..

sample_data

student_ID-6cf684e3-0088-4758-...

+ Code + Text

How to use (You can find the instruction with screenshots on homework slides)

1. Save a copy of this Colab notebook.
2. Download the JSON file you upload to NTU COOL.
3. Upload the JSON file to Colab.
4. Fill in your student ID (case insensitive) in the first cell.
5. Run both cells.
6. You should see the score you will get. Otherwise, there are some problem in your result file.

Your file should appear in the side panel

```
[ ]
import json
import os
import re

!git clone https://github.com/CheeEn-Yu/GenAI-Hw5.git --quiet

with open("GenAI-Hw5/Tang_testing_data.json", "r", encoding = "utf-8") as f:
    testData = json.load(f)

with open("GenAI-Hw5/Tang_testing_gt.txt", "r", encoding = "utf-8") as f:
    gts = f.readlines()

files = [f for f in os.listdir(".") if (re.match(f"{student_ID}(-[-\w]+)?.json", f, flags = re.IGNORECASE))]

if (len(files) == 1):
    print(f"Uploaded file: {files[0]}")
    with open(f"{files[0]}", "r") as f:
```

Disk 83.32 GB available

Connected to Python 3 Google Compute Engine backend

Copy of GenAI_HW5_Score_Parsing.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

..

sample_data

student_ID-6cf684e3-0088-4758-...

Disk 83.32 GB available

+ Code + Text

How to use (You can find the instruction with screenshots on homework slides)

1. Save a copy of this Colab notebook.
2. Download the JSON file you upload to NTU COOL.
3. Upload the JSON file to Colab.
4. Fill in your student ID (case insensitive) in the first cell.
5. Run both cells.
6. You should see the score you will get. Otherwise, there are some problem in your result file.

```
[ ]  student_ID = "student_ID" # case insensitive
```

Fill in your student ID in the first cell.
(Don't copy from the file name so that you can also check whether the ID in file name is correct)

```
[ ]
    testdata = json.load(f)

with open("GenAI-Hw5/Tang_testing_gt.txt", "r", encoding = "utf-8") as f:
    gts = f.readlines()

files = [f for f in os.listdir(".") if (re.match(f"{student_ID}(-[-\w]+)?.json", f, flags = re.IGNORECASE))

if (len(files) == 1):
    print(f"Uploaded file: {files[0]}")
    with open(f"{files[0]}", "r") as f:
        history = json.load(f)["history"]
```

Connected to Python 3 Google Compute Engine backend

Copy of GenAI_HW5_Score_Parsing.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

..

sample_data

student_ID-6cf684e3-0088-4758-...

+ Code + Text

How to use (You can find the instruction with screenshots on homework slides)

1. Save a copy of this Colab notebook.
2. Download the JSON file you upload to NTU COOL.
3. Upload the JSON file to Colab.
4. Fill in your student ID (case insensitive) in the first cell.
5. Run both cells.
6. You should see the score you will get. Otherwise, there are some problem in your result file.

[1]

```
1 stu
```

Run both cells by clicking the play button

```
1 imp
2 import os
3 import re
4
5 !git clone https://github.com/CheeEn-Yu/GenAI-Hw5.git --quiet
6
7 with open("GenAI-Hw5/Tang_testing_data.json", "r", encoding = "utf-8") as f:
8     testData = json.load(f)
9
10 with open("GenAI-Hw5/Tang_testing_gt.txt", "r", encoding = "utf-8") as f:
11     gts = f.readlines()
12
13 files = [f for f in os.listdir(".") if (re.match(f"{student_ID}(-[-\w]+)?.json", f, flags = re.IGNORECA
14
15 if (len(files) == 1):
16     print(f"Uploaded file: {files[0]}")
17     with open(f"{files[0]}", "r") as f:
18         history = json.load(f)["history"]
--NORMAL--
```

Disk 83.31 GB available

+ Code + Text

```python

    for sentence in gt.split("\t"):
        assert (sentence.strip() not in poem), f"{SEP}Your poem contains sentences that are in the original poem, please make sure

try:
    finalScores = json.loads(evaluationResult.split("```")[1])
except:
    raise ValueError(f"{SEP}Can not parse your final scores")

assert (len(finalScores) == 15), f"{SEP}Final scores do not contain score for each poem"

for (k, v) in finalScores.items():
    if (v["content"] >= 7.0): cnt["content"] += 1
    if (v["format"] >= 8.0): cnt["format"] += 1

for part in ["content", "format"]:
    if (cnt[part] >= 10): print(f"For {part} you get 5%")
    elif (cnt[part] >= 8): print(f"For {part} you get 4%")
    elif (cnt[part] >= 6): print(f"For {part} you get 3%")
    elif (cnt[part] >= 4): print(f"For {part} you get 2%")
    elif (cnt[part] >= 2): print(f"For {part} you get 1%")
    else: print(f"For {part} you get 0%")
```

--NORMAL--

```
For content you get 5%
For format you get 5%
```

If you see this, it means your submission file is fine, and this should be the credits you'll get

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-7-4dacb6fdea54> in <cell line: 22>()
     24
     25     for sentence in gt.split("\t"):
---> 26         assert (sentence.strip() not in poem), f"{SEP}Your poem contains sentences that are in the original poem, please make sure y
     27
     28 try:

AssertionError:
---------------------------------------------------------------------------
Your poem contains sentences that are in the original poem, please make sure your poems are created by the fine-tuned model
```

If you see something like these, it means there are some problem in your submission file.
If you cannot figure out what problem it is, please reach out to TAs.

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-8-4dacb6fdea54> in <cell line: 22>()
     21
     22 for (i, (poem, gt)) in enumerate(zip(poems, gts)):
---> 23     assert poem.startswith(f"{i + 1}. {testData[i]['input']}"), f"{SEP}Poems you submit does not match our test data"
     24
     25     for sentence in gt.split("\t"):

AssertionError:
---------------------------------------------------------------------------
Poems you submit does not match our test data
```

If You Have Any Questions

- NTU Cool HW5 作業討論區
  - 如果同學的問題不涉及作業答案或隱私，請**一律使用**NTU Cool 討論區
  - 助教們會優先回答NTU Cool討論區上的問題
- Email: ntu-gen-ai-2024-spring-ta@googlegroups.com
  - Title should start with [GenAI 2024 Spring Hw5]
  - Email with the wrong title will be moved to trash automatically
- TA Hours
  - Time: 3/29 (Fri.) 16:30 ~ 17:20, 4/5 (Fri.) 14:20 ~ 16:20 (online meet)
  - Location: 綜合大講堂

Appendix - How to activate GPU

- A GPU(Graphics Processing Unit) is a hardware component designed to accelerate the process of training models.
- Ensure that it is activated prior to commencing the training process

LLM_finetuning_GenAI.ipynb

File Edit View Insert Runtime Tools Help All changes saved

1

+ Code + Text

Comment Share

Connect Colab AI

1 Start codin

--NORMAL--

Run all Ctrl+Alt+R
Run before Ctrl+Alt+A
Run the focused cell Ctrl+Enter
Run selection
Run after Ctrl+Alt+B

Interrupt execution Ctrl+Alt+Shift+C
Restart session Ctrl+Alt+.
Restart session and run all
Disconnect and delete runtime

2

Change runtime type

Manage sessions
View resources
View runtime logs

Change runtime type

Runtime type

Python 3

3

Hardware accelerator

- CPU
- T4 GPU
- A100 GPU
- V100 GPU
- TPU

Want access to premium GPUs? Purchase additional compute units

Cancel Save

Appendix - Choose adapter on different steps

- Feel free to select the LoRA adapter at various stages of training and observe the variance in model output.
- In sample code, we opt for the adapter from the last steps.

Appendix - Friendly Reminder

This assignment marks the first attempt at training models for this course. During the training process, there may be encounters with bugs or poor training outcomes.

However, as the teacher mentioned in the first class, it is hoped that these negative experiences can serve as side effects of a vaccine, aiding all students in facing greater challenges in the future after completing this training assignment.

Other Ref.

[2106.09685] LoRA: Low-Rank Adaptation of Large Language Models

Taide model

聯發科模型