# GenAI-ML HW1

TA: 陳竣瑋 蔡昀劭 袁紹翔

ntu-gen-ai-ml-2025-fall-ta@googlegroups.com

Deadline: 2025/**10/17** 23:59:59 (UTC+8)

# Outline

- NTU COOL
- Tasks Overviews
- Hugging Face
- Colab
- Gradio
- TODO
- Submission and Grading

# Links

- Hugging Face: https://huggingface.co/
- Gemma-3-1b-it: https://huggingface.co/google/gemma-3-1b-it
- Colab: https://colab.research.google.com/drive/1PjLpwbhy8A6AMugfCJZGVZwjacBfszgk?usp=sharing

# Goals of This Homework

- Be familiar with NTU cool and google colab.
- Understand the concepts of **tokens, tokenizers, prompting, and chat templates.**
- You will see how the model behave with different prompt setting.
- You will learn how to build some simple user interface with **Gradio.**

# NTU COOL

# NTU COOL


帳戶
資訊總覽
課程
行事曆
收件匣
選課意願
FAQ
生成式人工智慧與機器學習導論 (AIA1390) › 線上測驗
114-1 (2025 Fall)
首頁
課程資訊
課程內容
公告
作業
線上測驗
討論
成績
成員
文件
Leganto
搜尋測驗
作業測驗
HW1
關閉時間 10月 17 at 23:59 截止時間 10月17日 下午 11:59 10 分 11 個問題

# NTU COOL - conti

## 說明

There are **11 questions** in the test.

The question and answer has two versions: one in English and one in Chinese.

Each question has **only one** correct answer.

Please use the provided Colab and what you have learned to choose the correct answer.

Remember to check that you have **finished all the questions and press the submit button.**

本測驗共有 **11題**。

題目和選項有兩個版本：一個是英文版，一個是中文版。

每題**只有一個**正確答案。

請使用提供的 Colab 以及你所學到的知識來選擇正確答案。

記得**確認已完成所有題目，並按下提交按鈕**。

**請勿同時使用多個設備、瀏覽器、分頁開啟同一份測驗的作答畫面，避免作答結果無法正確儲存。**

參加測驗

# NTU COOL - conti

- The quiz is in two language version

## 測驗說明

There are **11 questions** in the test.

The question and answer has two versions: one in English and one in Chinese.

Each question has **only one** correct answer.

Please use the provided Colab and what you have learned to choose the correct answer.

Remember to check that you have **finished all the questions and press the submit button.**

本測驗共有 **11題**。

題目和選項有兩個版本：一個是英文版，一個是中文版。

每題**只有一個**正確答案。

請使用提供的 Colab 以及你所學到的知識來選擇正確答案。

記得**確認已完成所有題目，並按下提交按鈕**。

# NTU COOL - conti

- Question Format:

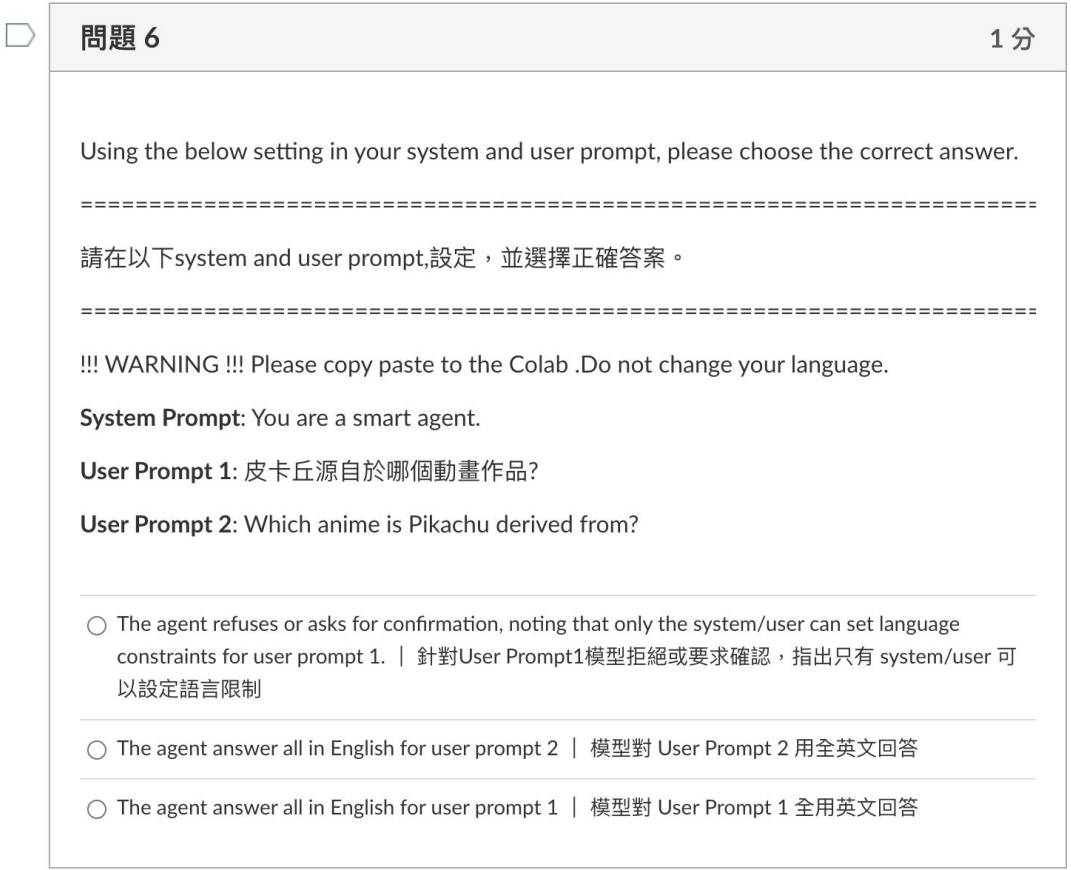
  English Question

  ==========

  中文問題

- Answer Format:

  English | 中文



問題 6 1 分

Using the below setting in your system and user prompt, please choose the correct answer.

==========

請在以下system and user prompt,設定，並選擇正確答案。

==========

!!! WARNING !!! Please copy paste to the Colab .Do not change your language.

**System Prompt**: You are a smart agent.

**User Prompt 1**: 皮卡丘源自於哪個動畫作品?

**User Prompt 2**: Which anime is Pikachu derived from?

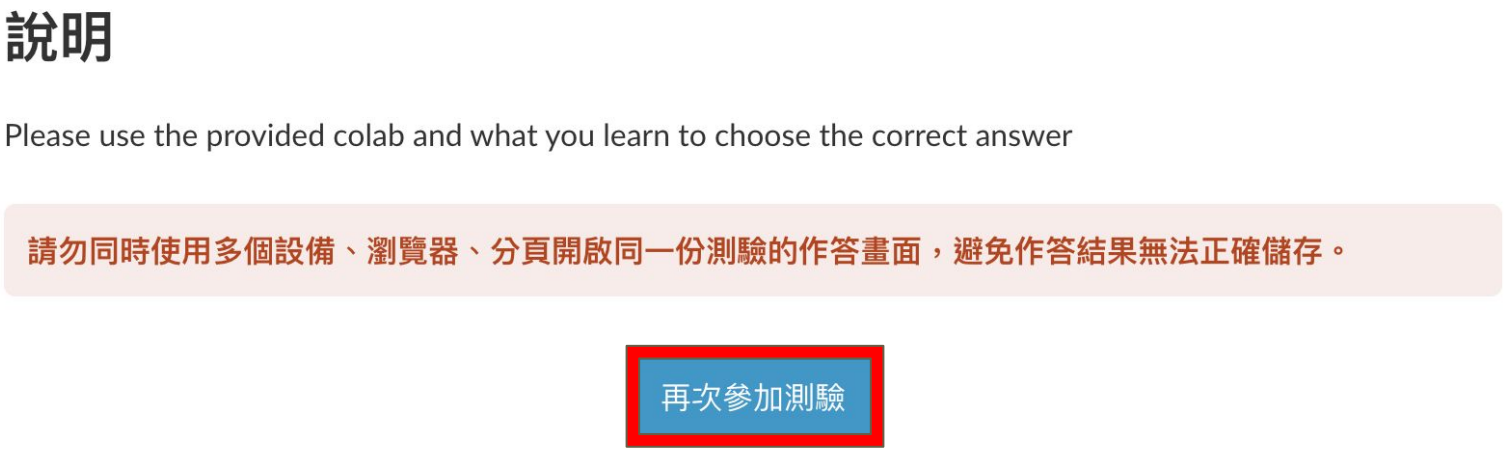
- ○ The agent refuses or asks for confirmation, noting that only the system/user can set language constraints for user prompt 1. | 針對User Prompt1模型拒絕或要求確認，指出只有 system/user 可以設定語言限制
- ○ The agent answer all in English for user prompt 2 | 模型對 User Prompt 2 用全英文回答
- ○ The agent answer all in English for user prompt 1 | 模型對 User Prompt 1 全用英文回答

# NTU COOL - conti

- Press this button if you want to submit your test

# NTU COOL - conti

- Press the button if you want to redo the test



# Tasks Overviews

# Part1 - Understanding Tokens in Large Language Models (5%)

- **Task Descriptions:**

  Learn about token information and usage in LLM

- **Q1: What is the vocabulary size of the Gemma-3-1B tokenizer? (1%)**
- **Q2: With the Gemma-3-1B tokenizer, which single token ID yields an English token? (1%)**
- **Q3: Encode the string「作業一」to token IDs (Gemma-3-1B). (1%)**
- **Q4: Which pair correctly reports the longest decoded token string in the vocabulary (token_id, character_length)? (1%)**
- **Q5: Given the prefix「阿姆斯特朗旋風迴旋加速噴氣式阿姆斯特朗砲」, which single Chinese character is the model's most probable next token? (1%)**

# Part2 - System and User Prompt Engineering (3%)

- **Task Descriptions:**
  - Observations of response with different System / User Prompt
    - System prompt:  It defines how the model should respond and establishes rules or constraints for the conversation.
    - User prompt:  It is the direct input or question given by the user. It contains the request, context, or task the AI needs to generate a response for.
- **Q6:instruction following (1%)**

  System Prompt: You are a smart agent.

  User Prompt 1: 皮卡丘源自於哪個動畫作品?

  User Prompt 2: Which anime is Pikachu derived from?

# Part2 - conti

- **Q7:– restrictive system prompt (1%)**

  System Prompt: You can only answer: I don’t know.

  User: 皮卡丘源自於哪個動畫作品?

- **Q8: – language constraint (1%)**

  System: Answer in English only

  User: 皮卡丘源自於哪個動畫作品?

# Part3 - Multi-Turn Conversation Implementation (2%)

- **Task Descriptions:**

  Learn about token information and usage in LLM

- **Q9: In the model.generate() call below,**

  **which parameter affects sampling randomness**

  **(given do_sample=True)? (0.5%)**

```
out = model.generate(
    **prompt,
    max_length=50,
    do_sample=True,
    top_k=k,
    temperature=1.0,
    pad_token_id=tok.eos_token_id
)
```

- **Q10:Which role in message correctly preserves the model response history? (0.5%)**

```
messages = [
    {"role":"history","content":"Who am I?"},
    {"role":"user","content":"Who am I?"},
    {"role":"assistant","content":"Who am I?"},
    {"role":"response","content":"Who am I?"}
]
```

# Part3 - conti

- **Q11: Compare the chat history before and after you uncomment the code and choose the right answer. (1%)**

```
while True:
    # USER INPUT: In practice, this would come from user interface or API call
    # For educational purposes, we use a fixed message to demonstrate the concept
    user_prompt = "What day is Tomorrow?"

    # ADD USER MESSAGE: Append the new user input to conversation history
    # This step is crucial - without it, the AI would have no context of what user said
    messages.append({"role": "user", "content": user_prompt})

    # Add History in user dialogue
    # TODO: uncomment the below code
    # if counter == 1:
    #     user_prompt = "Today is Friday."
    #     messages = messages[:-1]
    #     messages.append({"role": "user", "content": user_prompt})

    print("User:", user_prompt)
    # AI RESPONSE GENERATION: Use the complete conversation history for context-aware generation
    # The pipeline processes the entire message array, not just the latest user input
    outputs = pipe(
        messages,                    # Full conversation history for context
        do_sample=False,
        max_new_tokens=256,          # Limit response length (new tokens only)
    )
```

# Hugging Face

# Introduction

**Hugging Face**

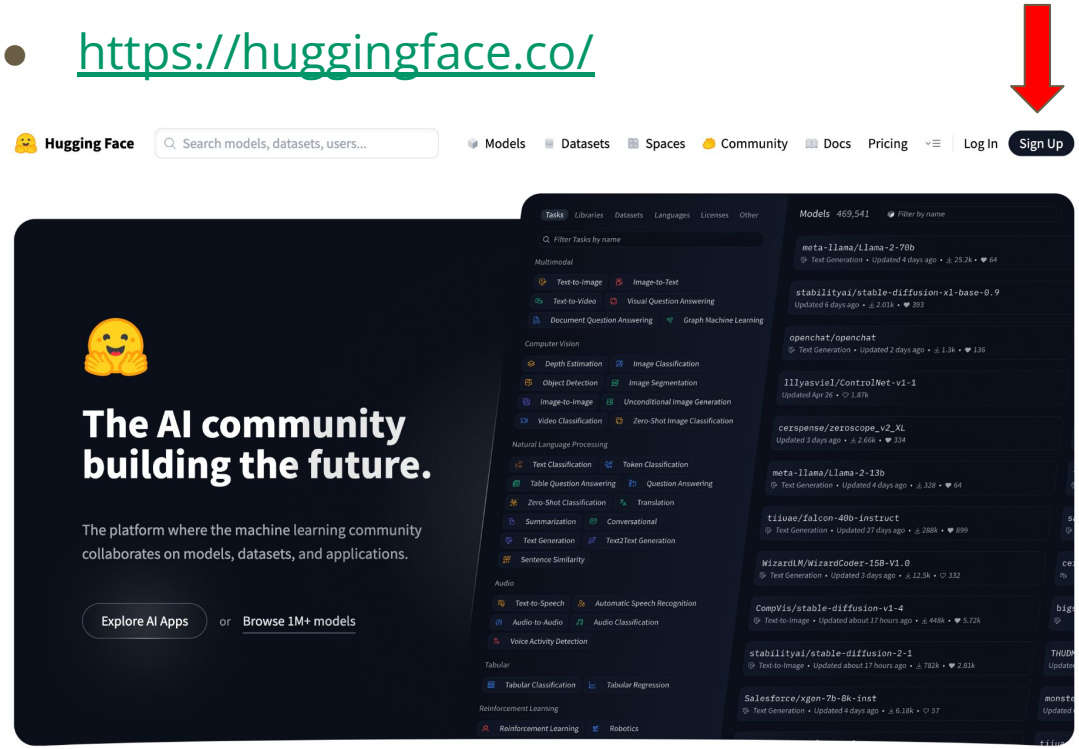
## What is Hugginface?

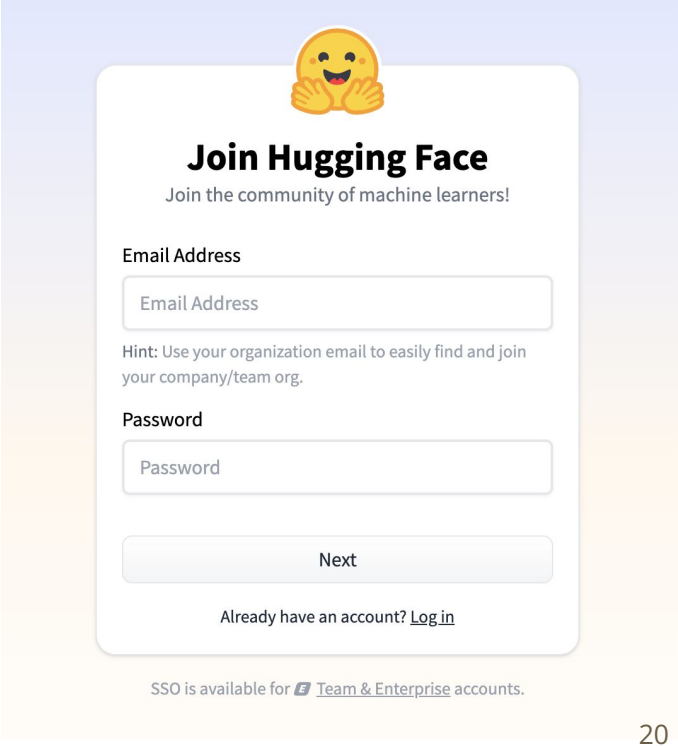
Hugging Face is an open platform for machine learning that lets you build, share, and collaborate on AI models, datasets, and applications.

- Zero setup for models – Use pre-trained models directly in your browser or via API
- Free community hub – Access thousands of open-source models, datasets, and spaces
- Easy sharing – Upload your own models or demos and share them instantly

# Sign up

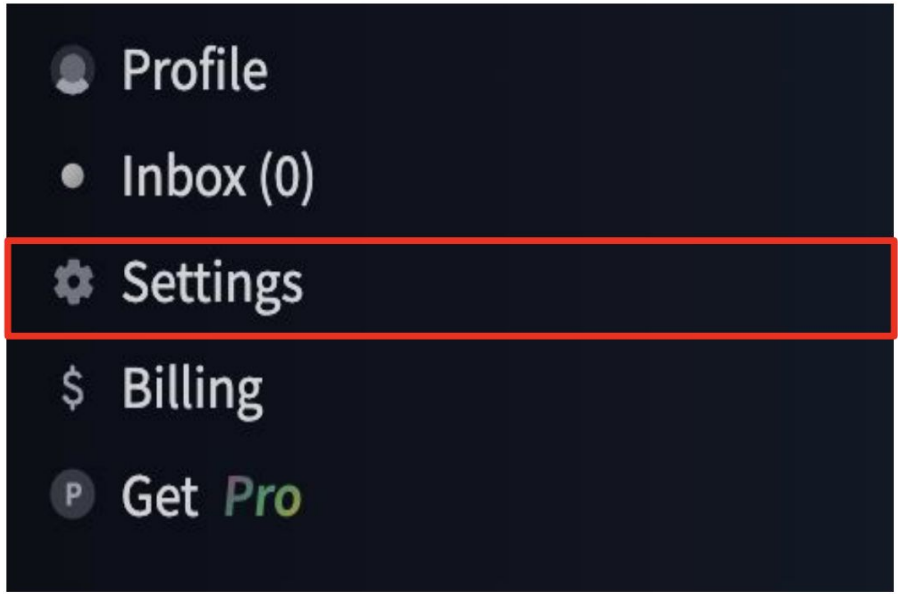
- https://huggingface.co/



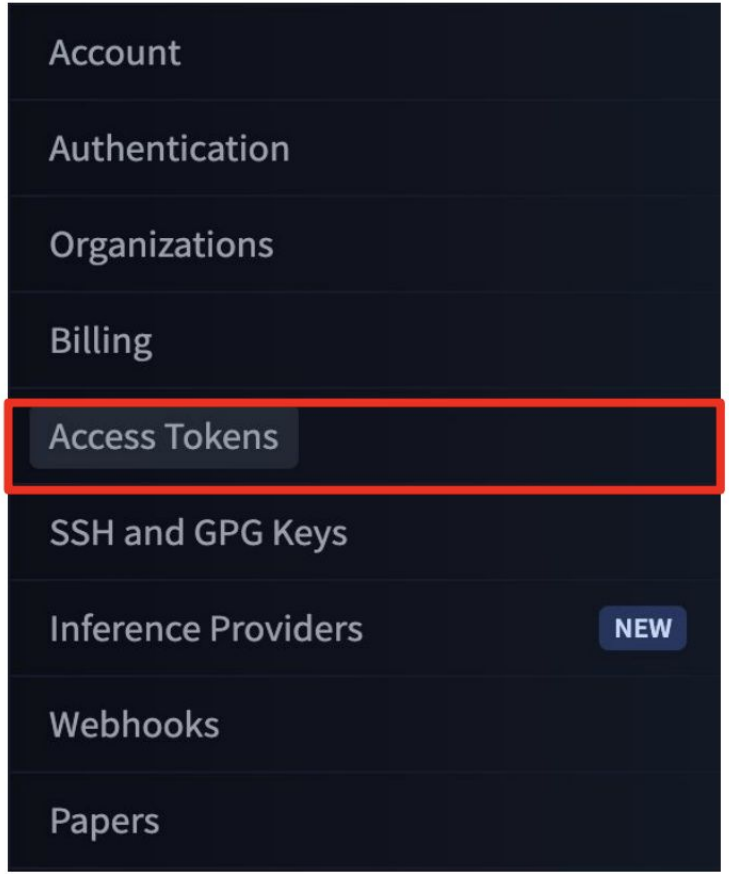
# Access Token 1/4

- Open your account settings



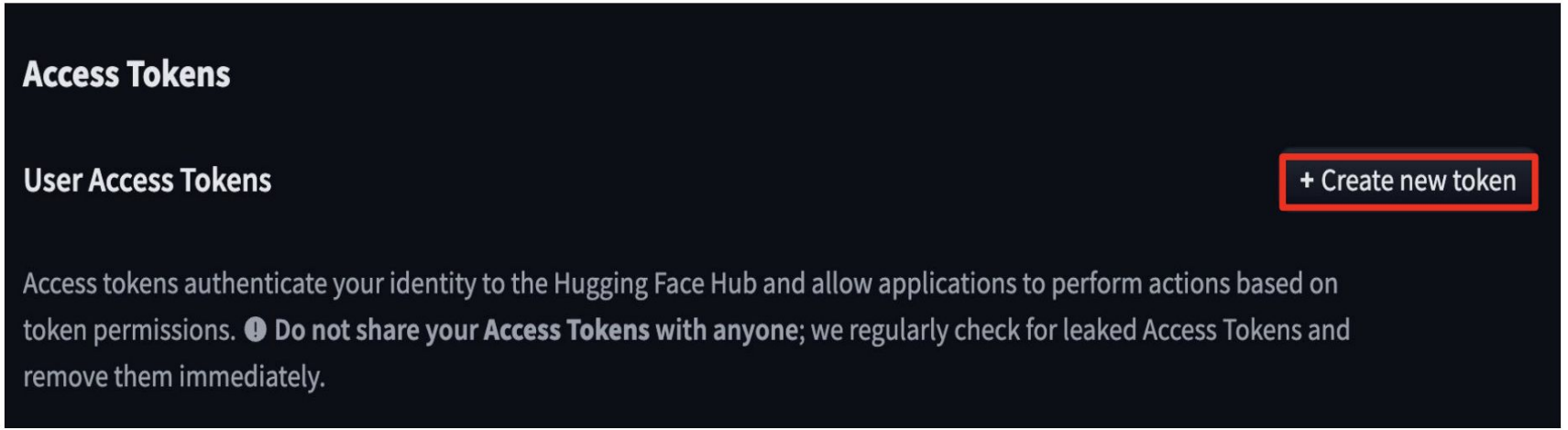
# Access Token - 2/4

- Click access tokens on the left bar



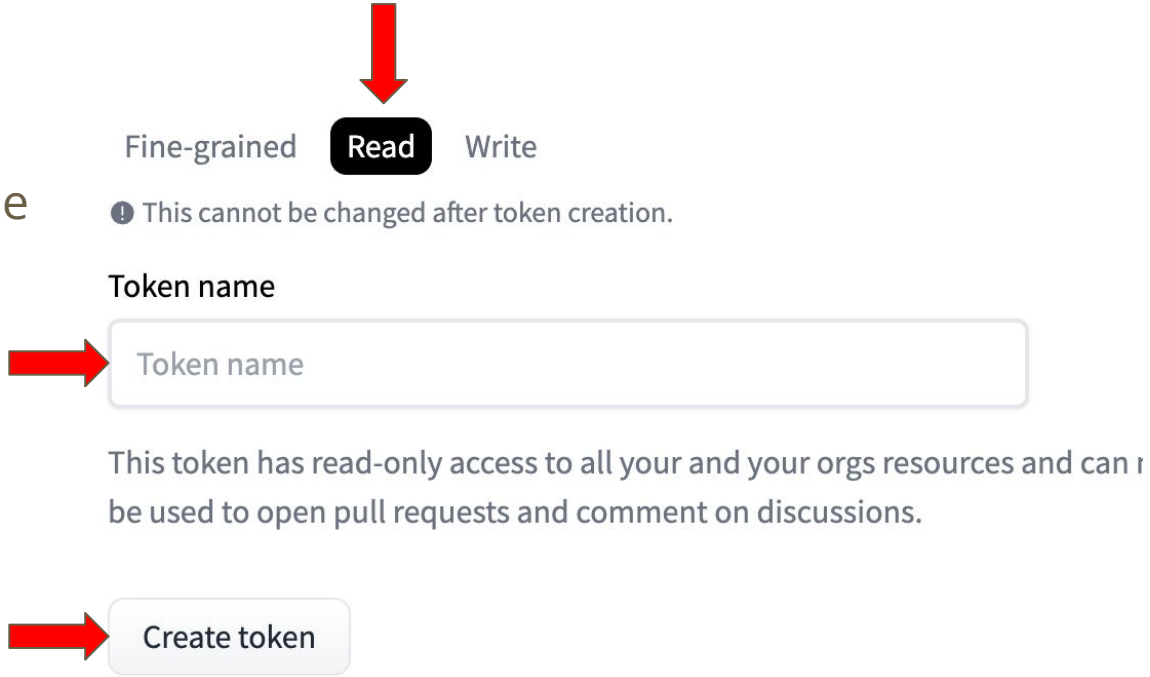
# Access Token - 3/4

- Click “Create new token”



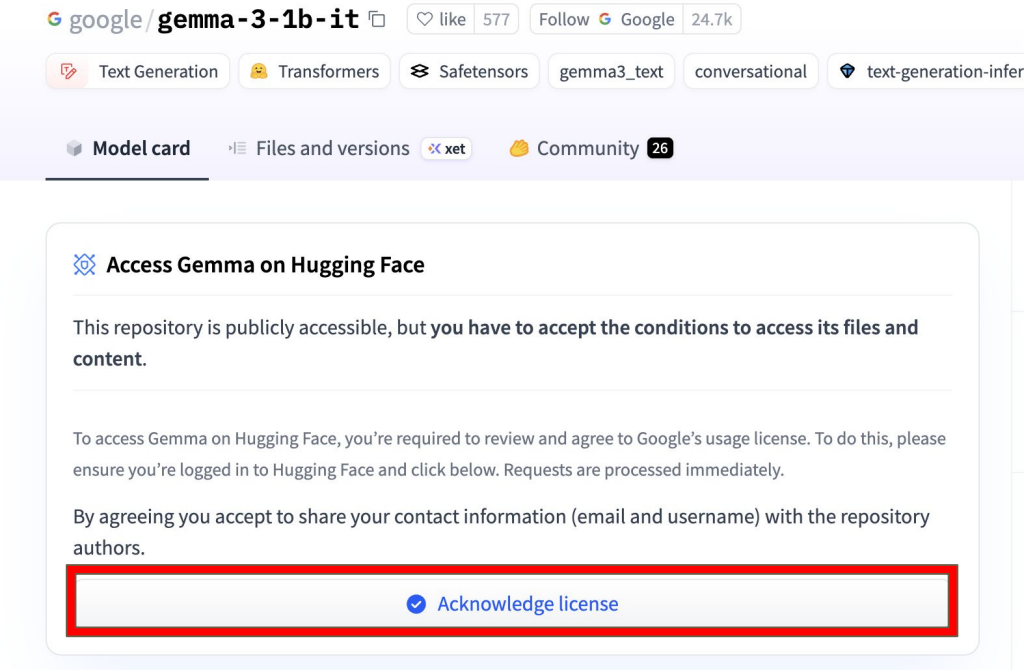
# Access Token - 4/4

- Select read token
- Enter your token name
- Create new token
- Copy the token



# Model permission - for hw1

- Click this link: https://huggingface.co/google/gemma-3-1b-it
- Enter your personal information



# Colab

# Introduction

## What is Colab?

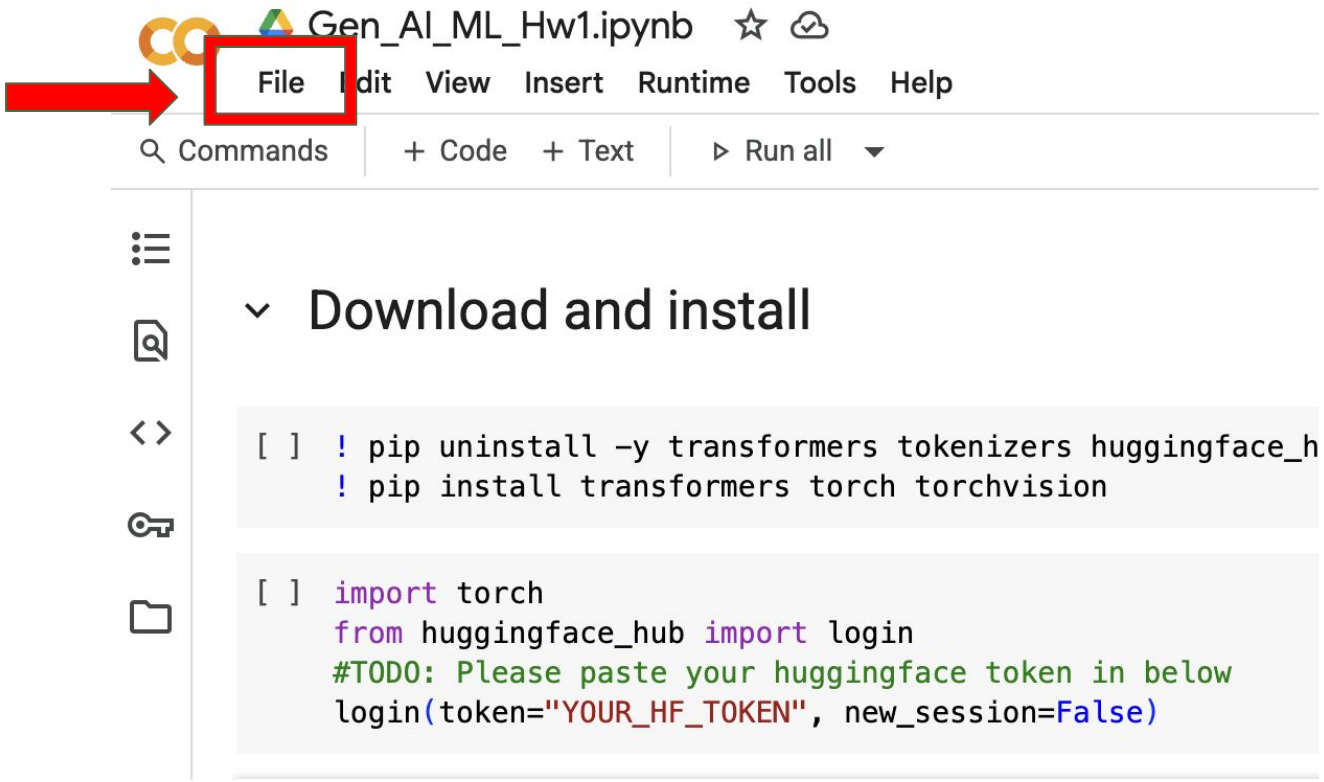
Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- **Zero configuration required**
- **Free access to GPUs**
- **Easy sharing**

# Colab code for Hw1

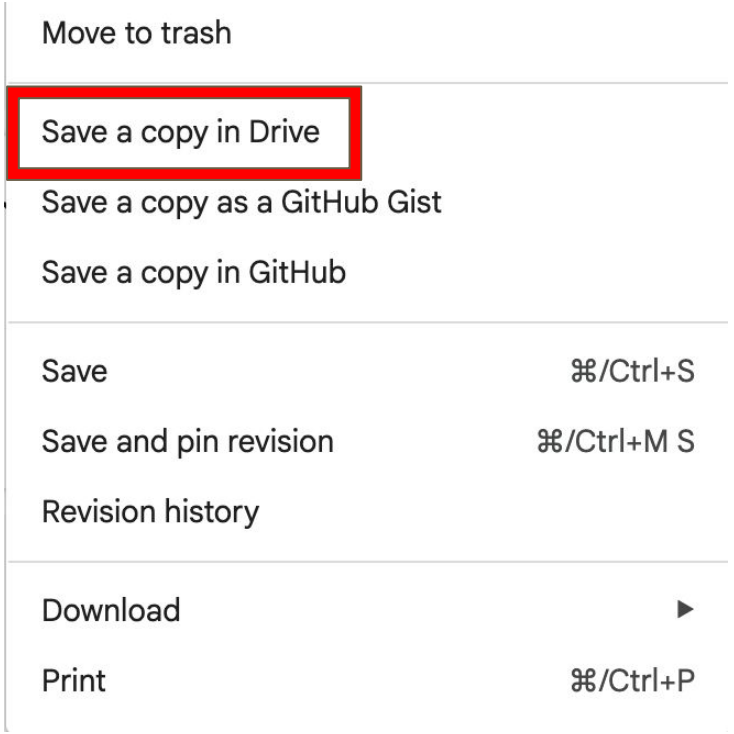
https://colab.research.google.com/drive/1PjLpwbhy8A6AMugfCJZGVZwjacBfszgk?usp=sharing

# Colab - conti


Gen_AI_ML_Hw1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Download and install
[ ] ! pip uninstall -y transformers tokenizers huggingface_h
! pip install transformers torch torchvision
[ ] import torch
from huggingface_hub import login
#TODO: Please paste your huggingface token in below
login(token="YOUR_HF_TOKEN", new_session=False)

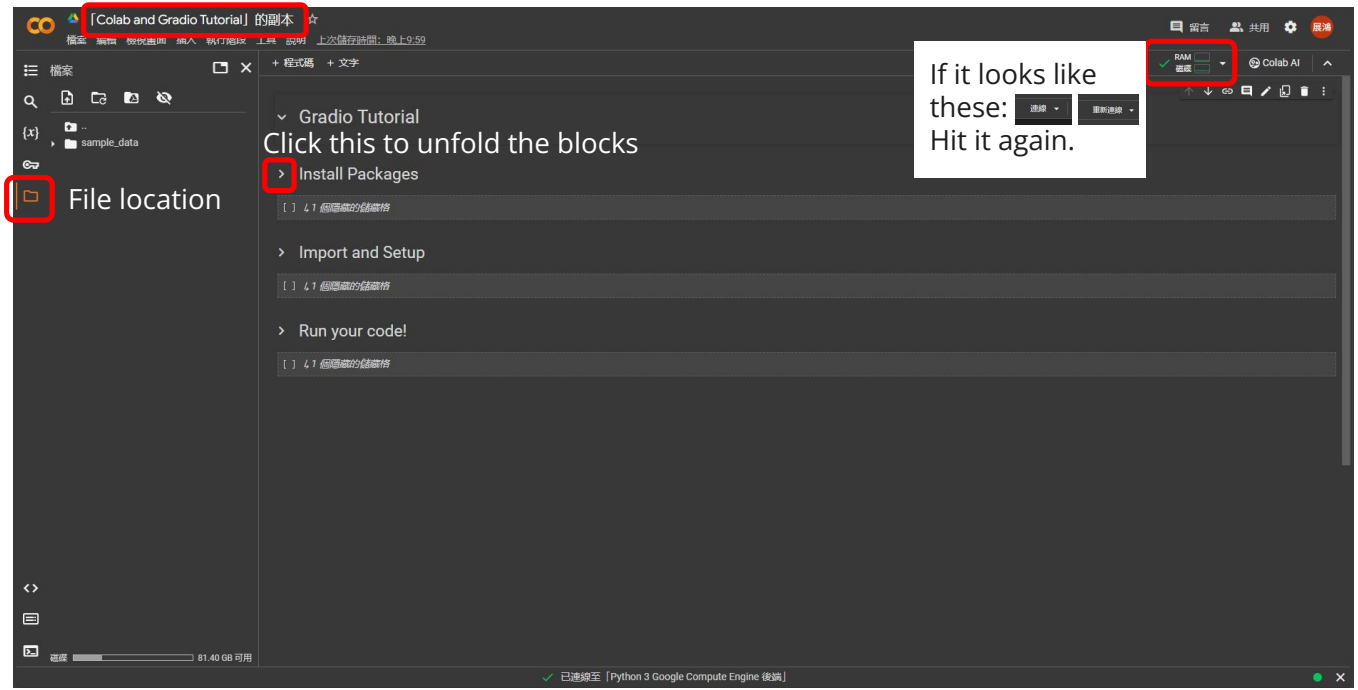
# Colab - conti

- Save a copy in drive



# Colab - conti

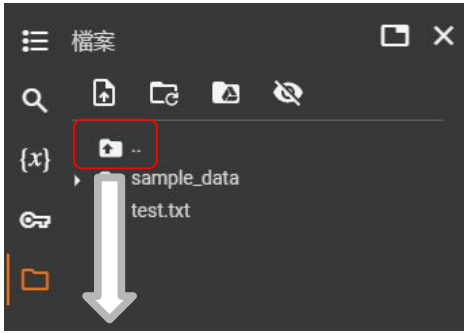
You can change the name here



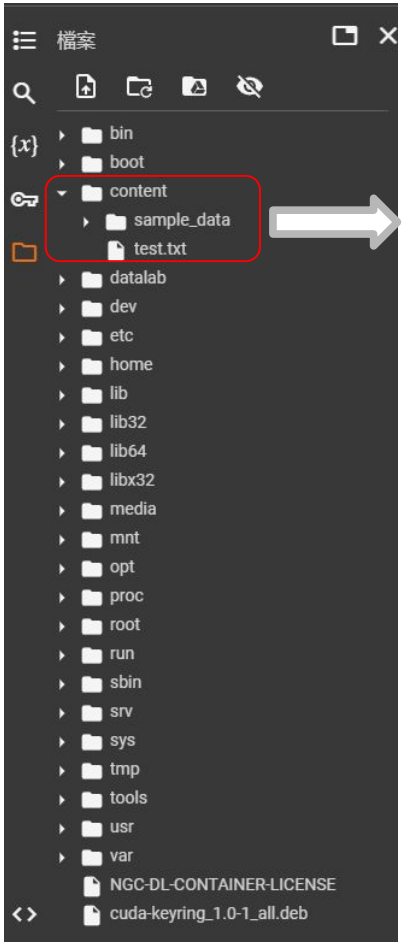
# Colab - conti


Upload your file

Download your file




If you accidently click this




You can find your files here

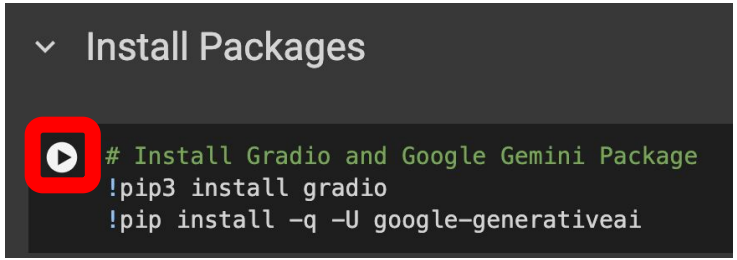
# Cells

Create a new cell by clicking on + Code or + Text

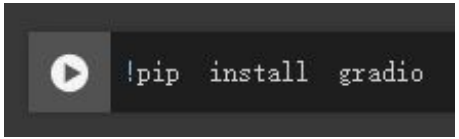
These options allow you to moving your cell up/down or copy and deleting it

# Run Cell

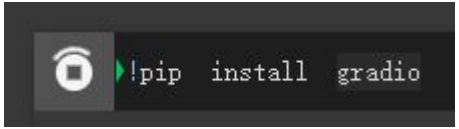
Run Cell by clicking on the 



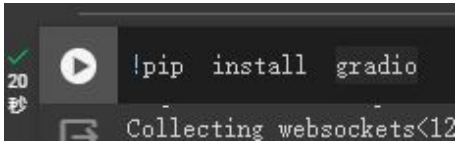

Code Unexecuted



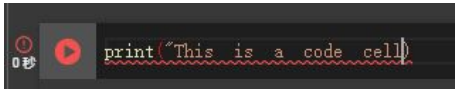

Code Executing



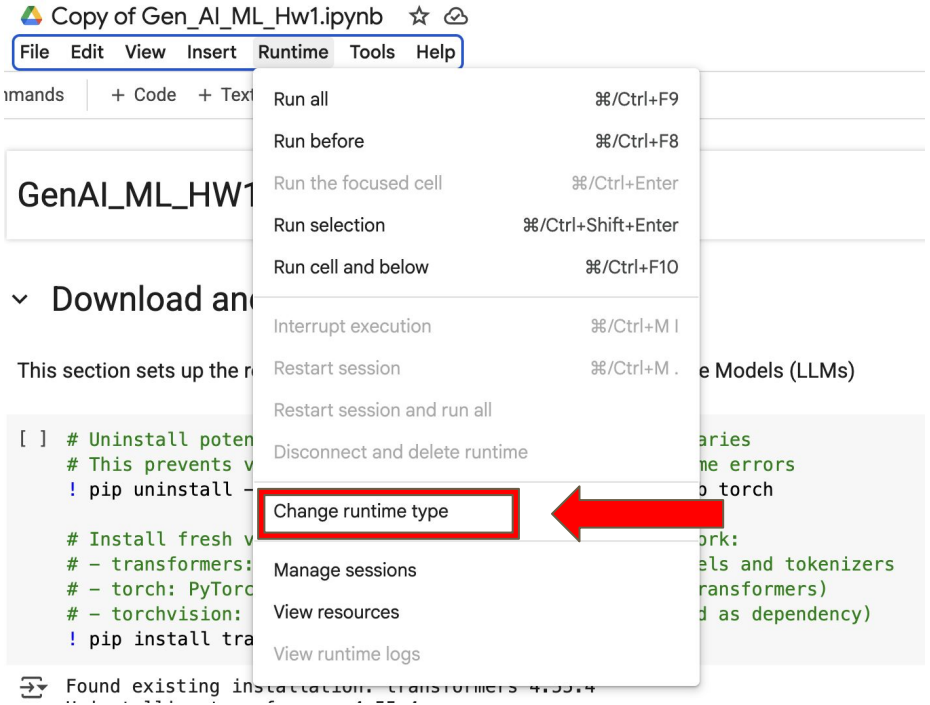

Code Executed

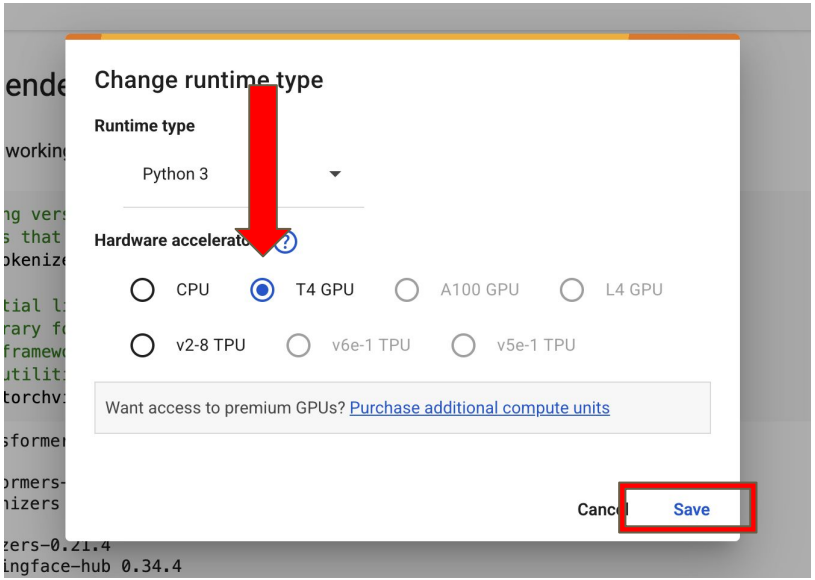



Error Occurred



# Make sure you are using GPU

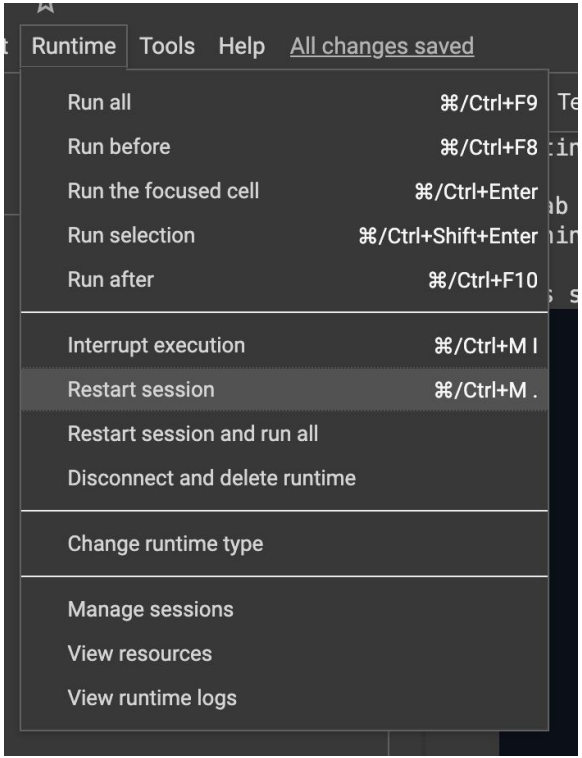


# Problems you may encounter

Colab will **automatically disconnect** if idle timeout (90 min., sometimes varying) or when your screen goes black.

If you “Disconnect and delete runtime” or “Restart Session”, your files will be gone.

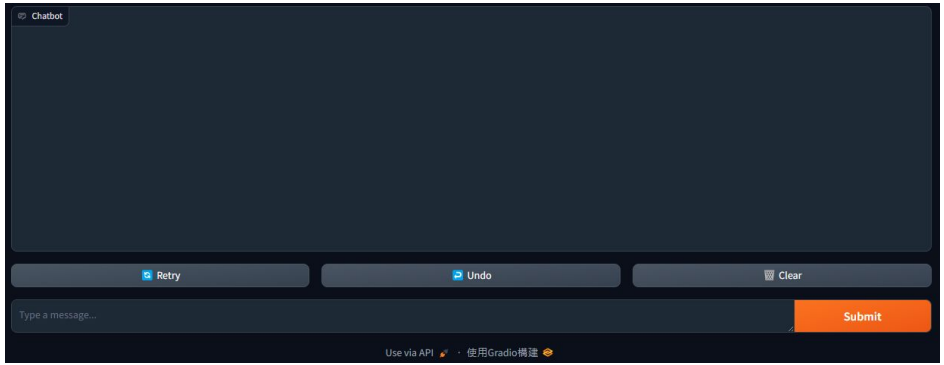


# Gradio

# Introduction

## What is Gradio?

Gradio is a tool that allows you to demo any application with a friendly website so that anyone can use it, anywhere.



# Launch the Hw1 gradio

- Run the cell in Part 4
- If you want to understand the meaning of the code, please check the command.

```
[ ]
# Required Libraries for Advanced Web Interface Development
# These imports provide the foundation for creating an interactive AI chatbot interface

import os, torch, transformers, gradio as gr
from transformers import (
    AutoModelForCausalLM,    # Core language model for text generation
    AutoTokenizer,           # Text tokenization and encoding utilities
)
import threading             # Multi-threading support for responsive UI (if needed)
```

```
[ ]
# Complete Interactive AI Chatbot Implementation with Gradio
# This comprehensive example demonstrates production-ready AI interface development

# ============================================================================
# MODEL SETUP AND CONFIGURATION
# ============================================================================

# Load the specified language model and tokenizer for the interface
LLM_NAME = "google/gemma-3-1b-it"  # Instruction-tuned Gemma model suitable for conversation
```

```
# ============================================================================
# CORE CONVERSATION FUNCTIONS
# ============================================================================

def format_chat_prompt(message, history):
    """
    Converts user input and conversation history into proper chat template format

    This function is crucial for maintaining conversation context and ensuring
    the model receives input in the format it was trained to expect.

    Args:
        message (str): Current user message/question
        history (list): List of (user_msg, assistant_msg) tuples from previous conversations
```

# Launch the Hw1 gradio - conti

- Copy and paste the link to your browser

```
# ============================================================================================
# INTERFACE LAUNCH: Deploy the application
# ============================================================================================

# Launch the Gradio interface with production-ready configuration
demo.launch(
    share=True,  # Create public URL for sharing (useful in Colab/cloud environments)
    debug=True   # Enable debug mode for development and learning
)
```

```
/tmp/ipython-input-3599748618.py:163: UserWarning: You have not specified a value for the `type` parameter. Defaulting to the
  chatbot = gr.Chatbot(
Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug=False in
* Running on public URL: https://791f9b58fc9c37e8f1.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the
```

🤖 大型語言模型聊天機器人

這是一個使用Transformers和Gradio建立的LLM聊天介面

對話記錄

生成參數

# Launch the Hw1 gradio - conti


Changeable parameters

Type your message here

Send message

Clear message

# TODO

# TODO

- Sign in **Hugging Face account and get access token**
- Get the **Gemma3 model permission** in Hugging Face
- Finish the sample code in **colab**

    a. Print out token information

    b. Get logits at the last position

    c. Modify different system / user prompt setting for observation

    d. Uncomment the messages setting for observation

- Answer the **multiple choice question** (only one right answer) in NTU COOL.

# Submission and Grading

# Submission & Deadline

- Submit your homework to **NTU Cool**
- 2025/**10/17** 23:59:59 (UTC+8)
- No late submission is allowed

# Grading Release Date

- The **total points** of this homework are **10 points**.
- The grading of the homework will be released before 2025/**10/20** 23:59:59 (UTC+8)

# If You Have Any Questions

- NTU Cool **HW1** 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-ml-2025-fall-ta@googlegroups.com
  - Title should start with [GenAI-ML 2025 Fall **HW1**]
  - Email with the wrong title will be moved to trash automatically
- TA Hours
  - Time:
    - **9/15, 9/22, 9/29, 10/6, 10/13** **Monday 20:00~22:00**
    - **9/19, 9/26, 10/3, 10/7, 10/17** **Friday 17:30~19:30**
  - Location: **Google meet**

# Appendix

# NTU COOL Quiz

**問題 1** 1 分

What is the vocabulary size of the Gemma-3-1B tokenizer?

==========

Gemma-3-1B 的 tokenizer 詞彙表大小是多少？

- ○ 128,486
- ○ 131,072
- ○ 320,357
- ○ 262,144

# NTU COOL Quiz - conti

**問題 2** 1 分

With the Gemma-3-1B tokenizer, which single token ID (decoded by itself) yields an English token?

==========

使用 Gemma-3-1B tokenizer，哪一個單一 token ID（單獨解碼時）會得到英文的 token？

- ○ 10,000
- ○ 80,000
- ○ 60,000
- ○ 20,000

# NTU COOL Quiz - conti

**問題 3** 1 分

Encode the string「**作業一**」to token IDs (Gemma-3-1B). Which is correct?

==========

將字串「**作業一**」編碼成 token IDs（Gemma-3-1B）。哪個是正確的？

- ○ [46306]
- ○ [46306, 237009]
- ○ [2746, 10552]
- ○ [237009, 46306]

# NTU COOL Quiz - conti

**問題 4** 1 分

Which pair correctly reports the longest decoded token string in the vocabulary (token_id, character_length)?

==========

哪一組 (token_id, 字元長度) 正確描述了詞彙表中最長的解碼字串？

- ○ (137, 31)
- ○ (4096, 37)
- ○ (512, 24)
- ○ (2048, 27)

# NTU COOL Quiz - conti

**問題 5** 1 分

Given the prefix 「阿姆斯特朗旋風迴旋加速噴氣式阿姆斯特朗砲」, which single Chinese character is the model's most probable next token?

==========

給定文字「阿姆斯特朗旋風迴旋加速噴氣式阿姆斯特朗砲」，模型最有可能產生的下一個中文字是什麼？

- ○ 砲
- ○ 超
- ○ 槍
- ○ 塔

# NTU COOL Quiz - conti

**問題 6** 1 分

Using the below setting in your system and user prompt, please choose the correct answer.

==========

請在以下system and user prompt,設定，並選擇正確答案。

==========

!!! WARNING !!! Please copy paste to the Colab .Do not change your language.

**System Prompt**: You are a smart agent.

**User Prompt 1**: 皮卡丘源自於哪個動畫作品?

**User Prompt 2**: Which anime is Pikachu derived from?

- ○ The agent refuses or asks for confirmation, noting that only the system/user can set language constraints for user prompt 1. | 針對User Prompt1模型拒絕或要求確認，指出只有 system/user 可以設定語言限制
- ○ The agent answers all in English for user prompt 1 | 模型對 User Prompt 1 全用英文回答
- ○ The agent answers all in English for user prompt 2 | 模型對 User Prompt 2 用全英文回答

# NTU COOL Quiz - conti

**問題 7** 1 分

Using the below setting in your system and user prompt, please choose the correct answer.

==========

請在以下system and user prompt,設定，並選擇正確答案。

==========

!!! WARNING !!! Please copy paste to the Colab .Do not change your language.

**System Prompt:** *You can only answer: I don't know.*

**User Prompt:** 皮卡丘源自於哪個動畫作品?

- ○ The agent answers the question not saying "I don't know". | 模型回答了問題，而不是「I don't know」
- ○ The agent answers "我不知道" | 模型回答「我不知道」
- ○ The agent answers "I don't know" | 模型回答「I don't know」
- ○ The agent outputs "I don't know" but adds extra words/symbols/translation (e.g., "I don't know, sorry.") | 模型輸出「I don't know」但加上額外文字 / 符號 / 翻譯（例如 "I don't know, sorry."）

# NTU COOL Quiz - conti

**問題 8** 1 分

Using the below setting in your system and user prompt, please choose the correct answer.

==========

請在以下system and user prompt,設定，並選擇正確答案。

==========

!!! WARNING !!! Please copy paste to the Colab .Do not change your language.

**System Prompt:** *Answer in English only*

**User Prompt:**皮卡丘源自於哪個動畫作品?

**!!! NOTE that there's no punctuation in system prompt.**

**!!! 請注意本題system prompt 沒有標點符號**

- ○ The agent replies with mixed Chinese and English | 模型混用中英文回答
- ○ The agent responds in English only to request the user to ask in English or refuses to answer the content.| 模型僅用英文回應，要求使用者改用英文詢問或拒答
- ○ The agents answer in Chinese.| 模型用中文回答
- ○ The agent answers in English.| 模型用英文回答

# NTU COOL Quiz - conti

**問題 9** 0.5 分

In the model.generate() call below, which parameter affects sampling randomness (given do_sample=True)?

==========

在以下 model.generate() 呼叫中，哪個參數會影響抽樣隨機性 (假設 do_sample=True)?

==========

```
out = model.generate(
    **prompt,
    max_length=50,
    do_sample=True,
    top_k=k,
    temperature=1.0,
    pad_token_id=tok.eos_token_id
)
```

- ○ max_length
- ○ attention_mask
- ○ top_k
- ○ pad_token_id

# NTU COOL Quiz - conti

**問題 10** 0.5 分

Which role in messages correctly preserves the model response history?

==========

在 message結構中，哪個 role 正確保存了模型的回應歷史？

==========

```
messages = [
    {"role":"history","content":"Who am I?"},
    {"role":"user","content":"Who am I?"},
    {"role":"assistant","content":"Who am I?"},
    {"role":"response","content":"Who am I?"}
]
```

- ○ history
- ○ assistant
- ○ response
- ○ user

# NTU COOL Quiz - conti

**問題 11** 1 分

Compare the chat history before and after you uncomment the code and choose the right answer.

==========

比較取消註解程式碼前後的聊天記錄，選出正確的答案。

==========

```
while True:
    # USER INPUT: In practice, this would come from user interface or API call
    # For educational purposes, we use a fixed message to demonstrate the concept
    user_prompt = "What day is Tomorrow?"

    # ADD USER MESSAGE: Append the new user input to conversation history
    # This step is crucial - without it, the AI would have no context of what user said
    messages.append({"role": "user", "content": user_prompt})

    # Add History in user dialogue
    # TODO: uncomment the below code
    # if counter == 1:
    #     user_prompt = "Today is Friday."
    #     messages = messages[:-1]
    #     messages.append({"role": "user", "content": user_prompt})

    print("User:", user_prompt)
    # AI RESPONSE GENERATION: Use the complete conversation history for context-aware generation
    # The pipeline processes the entire message array, not just the latest user input
    outputs = pipe(
        messages,                    # Full conversation history for context
        do_sample=False,
        max_new_tokens=256,         # Limit response length (new tokens only)
    )
```

- ○ The agent says tomorrow is Friday before uncommenting. | 模型反註解之前，表示明天是星期五。
- ○ The agent says tomorrow is Saturday after uncommenting. | 模型反註解之後，表示明天是星期六。
- ○ The agent says tomorrow is Saturday before uncommenting. | 模型反註解之前，表示明天是星期六。
- ○ The agent refused to answer after uncommenting. | 模型反註解之後，拒絕回答。