

大型語言模型

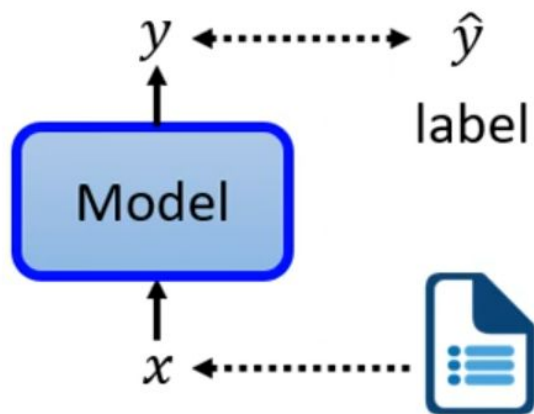
課程須知

助教：蔡明晏、林言翰、紀宇烜、葉志銓、范姜伯軒、陳冠榮

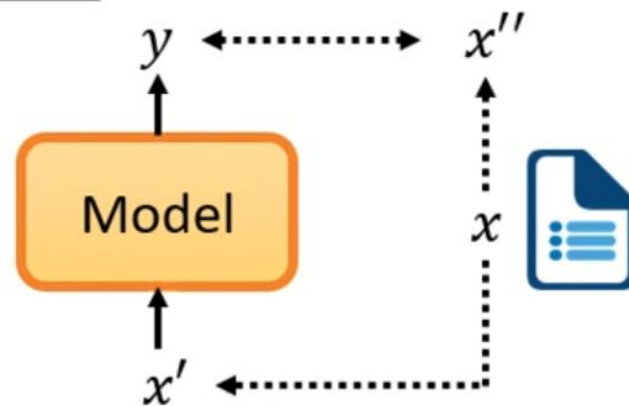
What is LLM ?

How to train LLM ?

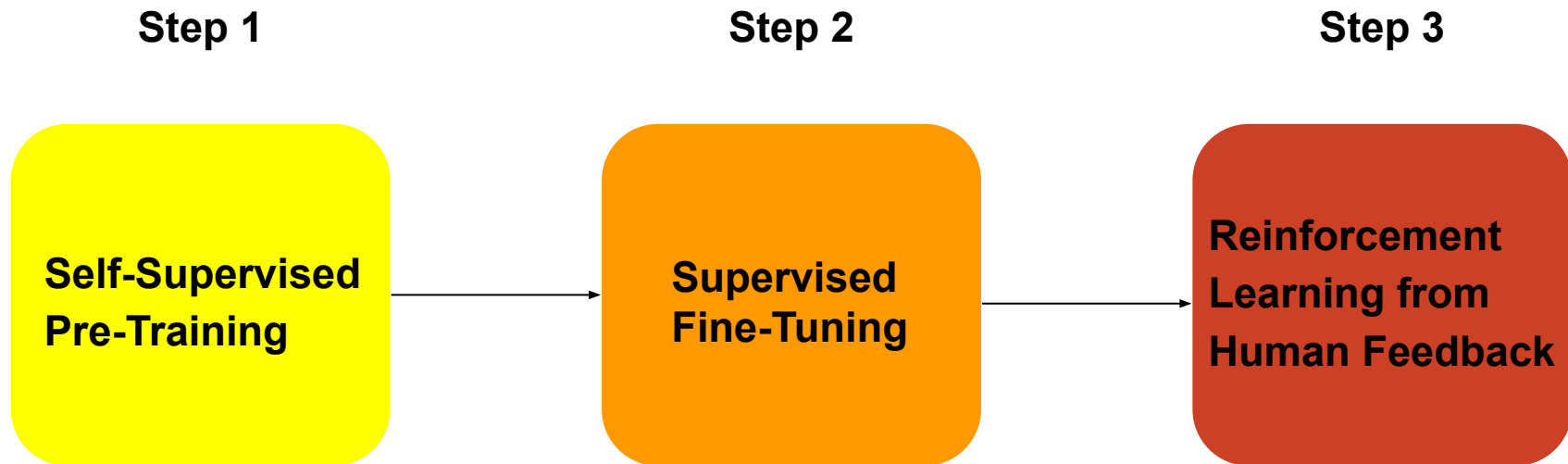
Supervised



Self-supervised

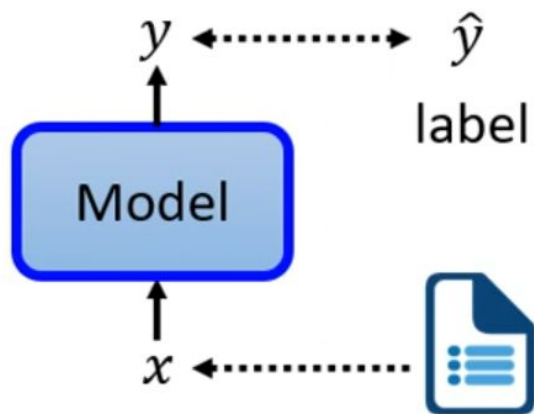


How to train LLM ?

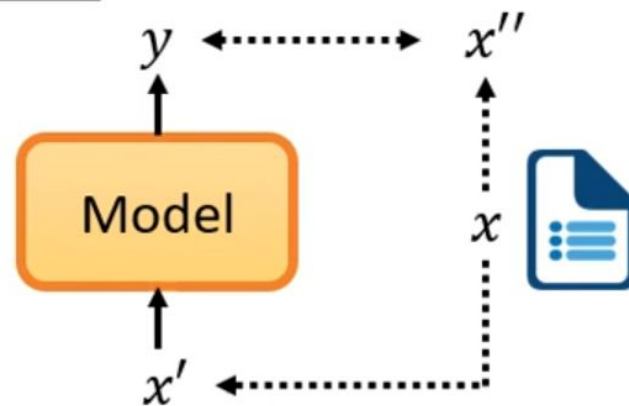


How to train model ?

Supervised



Self-supervised



大綱

- 評分方式
- 課程進度表
- 作業/專案規範
- 助教時間
- 聯絡方式
- 自學資源

評分方式

- Homework (or Task Competition) (60%)
- Final Project (40%)
- Class Attendance (+10%)

課程進度表

每週進度表		
週次	上課日期	課程進度、內容、主題
1	2025-09-05(五)	Feedforward and Convolutional Neural Networks
2	2025-09-12(五)	Regularization for Optimization in Deep Learning
3	2025-09-19(五)	Recurrent Neural Network and Sequential Learning
4	2025-09-26(五)	N-Gram Language Models and Topic-Based Language Models (Final project grouping)
5	2025-10-03(五)	RNN Language Models and Language Understanding (HW1 announcement)
6	2025-10-10(五)	Attention Networks and Transformers
7	2025-10-17(五)	State-Space Language Models
8	2025-10-24(五)	BERT Encoder and GPT Decoder
9	2025-10-31(五)	Retrieval, Augmentation and Generation (Final project proposal)
10	2025-11-07(五)	Generation with Prompting Strategies (HW2 announcement)
11	2025-11-14(五)	LLMs with GPT and LLaMA
12	2025-11-21(五)	LLM Model Trends and Generative AI
13	2025-11-28(五)	Final Presentation
14	2025-12-05(五)	AI Computing Architecture for LLM
15	2025-12-12(五)	Final Presentation
16	2025-12-19(五)	DaVinci GAI Platform and Applications

作業規範

- 使用Python撰寫
- 助教會幫忙釐清題意觀念，但不會幫忙 Debug。
- 需要繳交程式碼、結報。
- 繳交期限大約兩星期，可以補交，但分數會打折
- 禁止抄襲

專案規範

- 專題範圍為課程內容
- 3 - 4人一組
- 每組需要先繳交一份 proposal
- 助教會先針對各組提交的 proposal 給予實作上的建議讓同學參考
- 若提出的 proposal 在實作上有困難，跟助教討論後可調整 proposal 的內容，但
最晚須在 final project 報告前兩周寄信告知助教

助教時間

- 實體詢問

- 週三19:00-20:00 工程四館708
- 欲前往請提前寄信告知, 並描述問題 內容, 以利助教能更快速的協助解惑

- 線上詢問

- NewE3討論區優先, 其次為助教電子信箱 nycuml1ab@gmail.com

聯絡方式

- 實體
 - 工程四館708
- 電子信箱
 - nycumllab@gmail.com

自學資源

- [Kaggle](#)
- [Pytorch Tutorials](#)
- [Hugging Face](#)
- [iThome](#)
- [知乎](#)
- [scikit-learn](#)
- [NumPy Doc.](#)
- [Deep Learning.\(Goodfellow\)](#)

Google

colab





Google Colab


- 筆記本都會建置在Google Drive中




Google Colab

- 套件安裝 & 雲端存取





```
!pip install datasets
!pip install evaluate
!pip install transformers
```



```
Collecting datasets
  Downloading datasets-2.21.0-py3-none-any.whl
Requirement already satisfied: filelock in /usr/
Requirement already satisfied: numpy>=1.17 in
Collecting pyarrow>=15.0.0 (from datasets)
  Downloading pyarrow-17.0.0-cp310-cp310-many
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.met
Requirement already satisfied: pandas in /usr/
Requirement already satisfied: requests>=2.32.
Requirement already satisfied: tqdm>=4.66.3 in
Collecting xxhash (from datasets)
```



```
from google.colab import drive
drive.mount('/content/drive')
```

1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
17.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.7.4)

要允許這個筆記本存取你的 Google 雲端硬碟檔案嗎？

這個筆記本要求存取你的 Google 雲端硬碟檔案。獲得 Google 雲端硬碟存取權後，筆記本中執行的程式碼將可修改 Google 雲端硬碟的檔案。請務必在允許這項存取權前，謹慎審查筆記本中的程式碼。

不用了，謝謝

[連線至 Google 雲端硬碟](#)

✓
26
秒



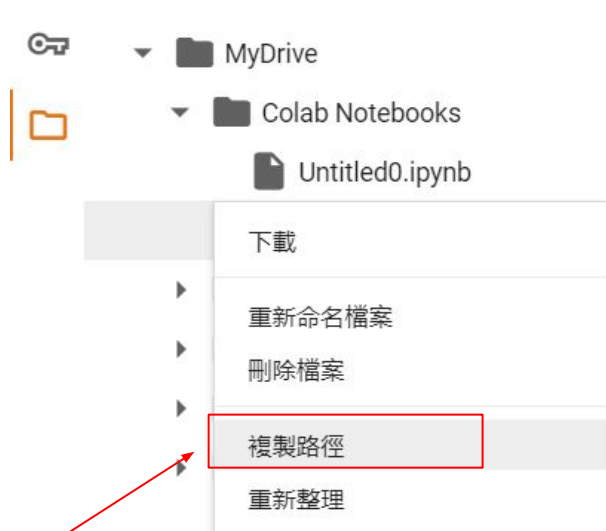
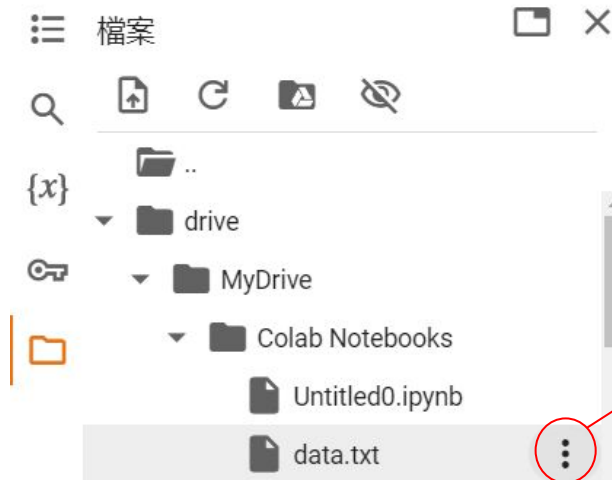
```
from google.colab import drive
drive.mount('/content/drive')
```



```
Mounted at /content/drive
```


Google Colab

- 載入資料集

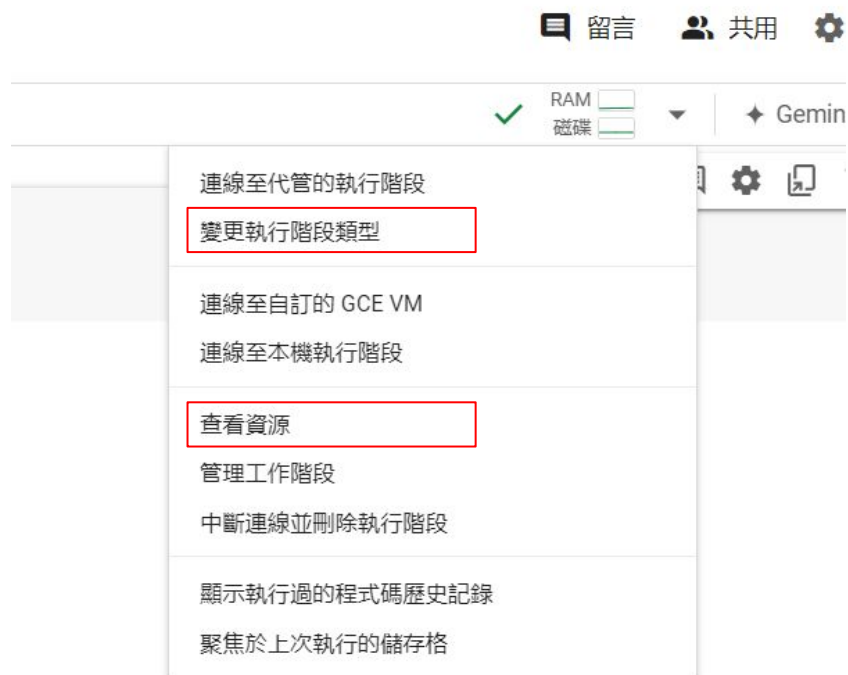


```
path = '/content/drive/MyDrive/Colab Notebooks/data.txt'
with open(path, 'r') as file:
    data = file.read()
    print(data)
```

Hello!!!

Google Colab

- 使用GPU



Google Colab

變更執行階段類型

執行階段類型

Python 3

硬體加速器 ?

- ☒ CPU ☐ T4 GPU ☐ A100 GPU ☐ L4 GPU
- ☐ TPU v2

想要使用付費 GPU 嗎? [購買額外運算單元](#)

取消

儲存



RAM
磁碟



◆ Gemini



資源 ×



你並未訂閱這項服務。 [瞭解詳情](#)

目前沒有可用的運算單元。本產品不保證免費版本的資源分配。如需購買更多運算單元，請按[這裡](#)。

根據目前的用量層級，這個執行階段最長可持續 42 小時 40 分鐘。

[管理工作階段](#)

需要更多記憶體和磁碟空間嗎? [升級至 Colab Pro](#) ×

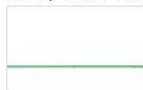
Python 3 Google Compute Engine 後端

目前顯示 8月24日 至 凌晨12:11 之間的資源

系統 RAM
1.5 / 12.7 GB



磁碟
31.1 / 107.7 GB

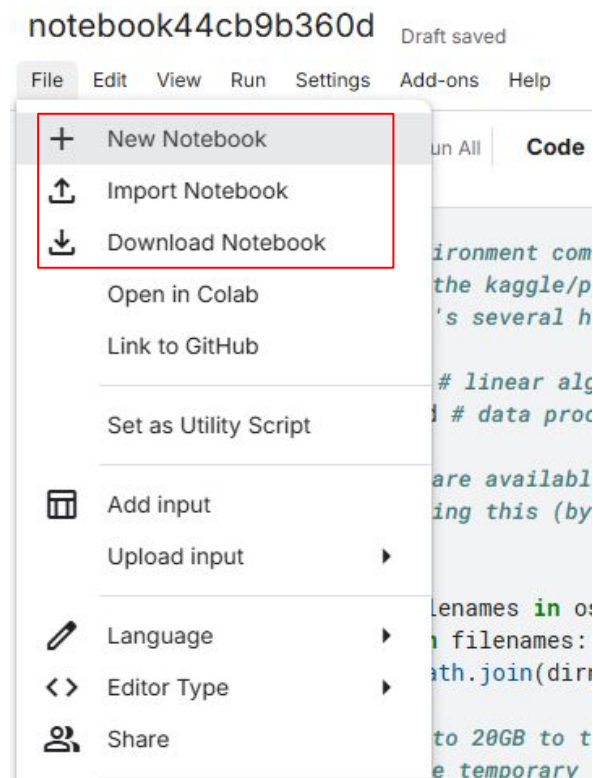


[變更執行階段類型](#)

kaggle

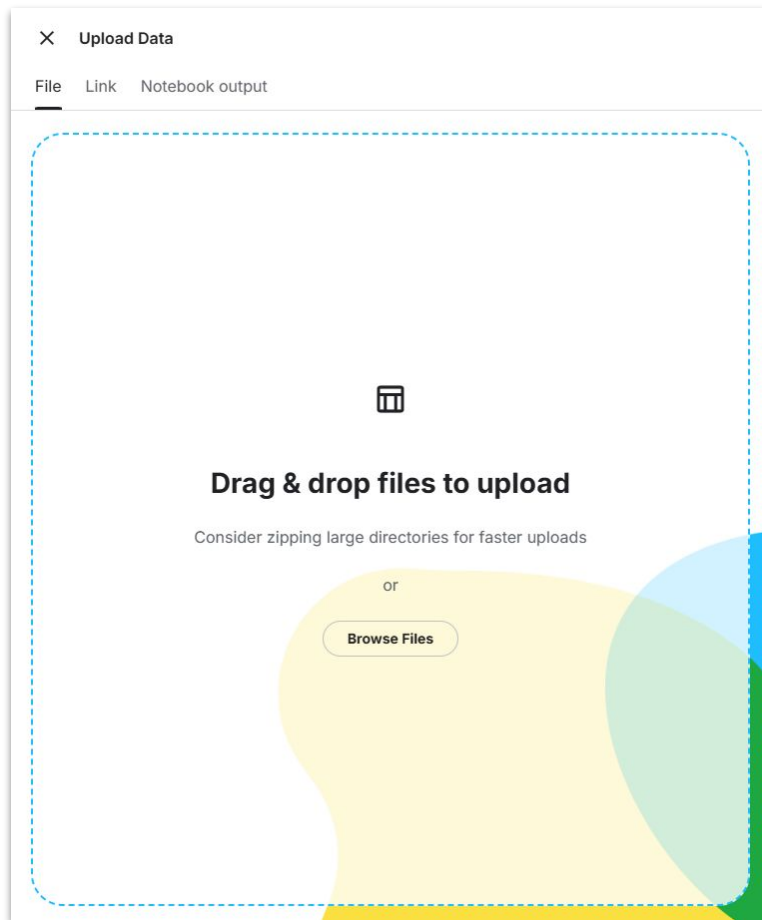
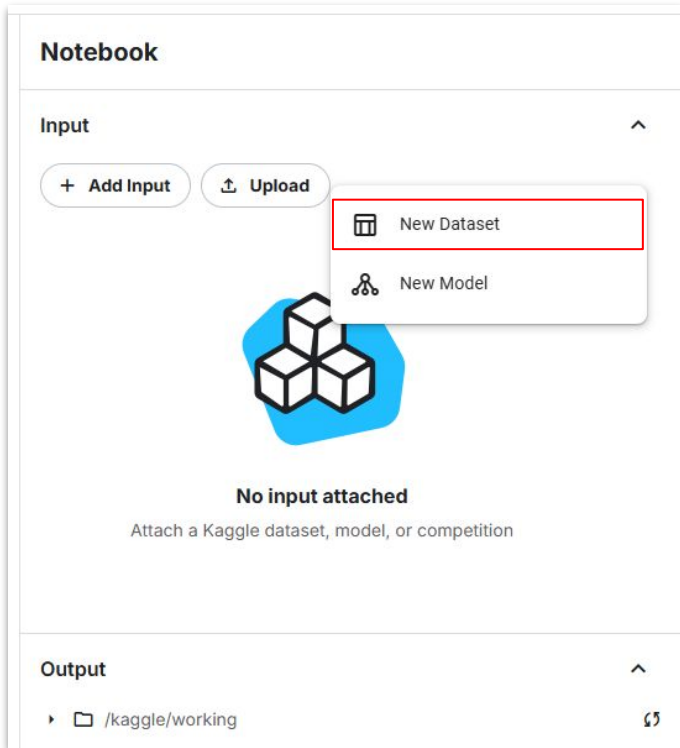
kaggle

- 新增 / 上傳筆記本



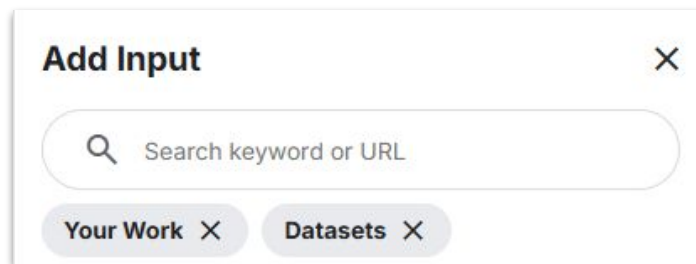
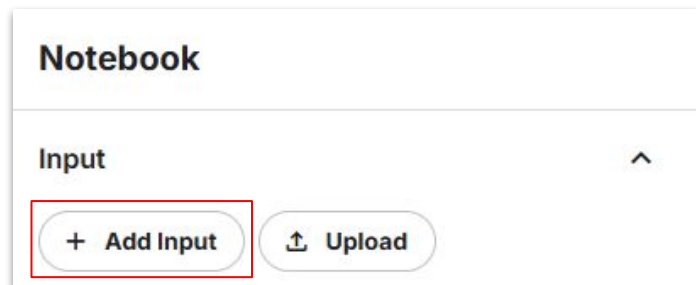
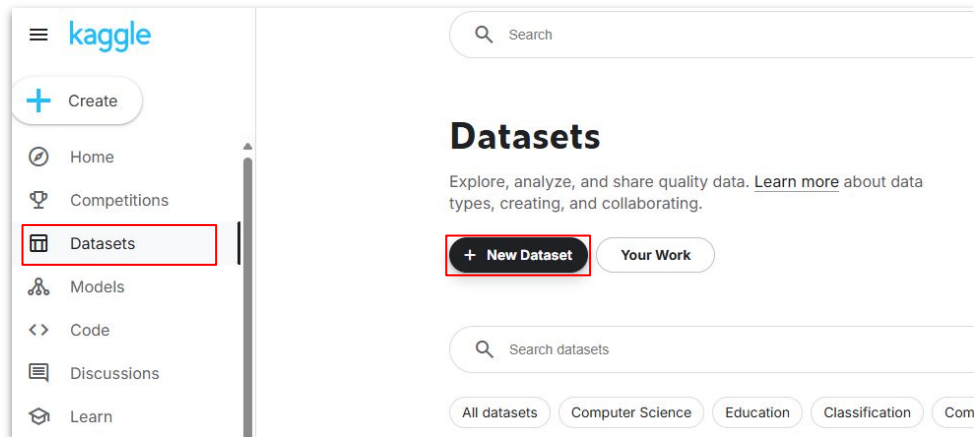
kaggle

- 上傳資料集 (方法一)



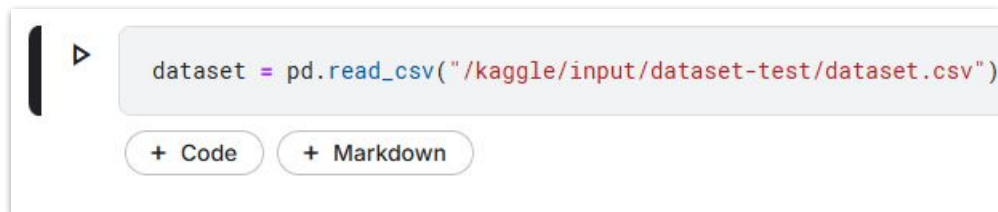
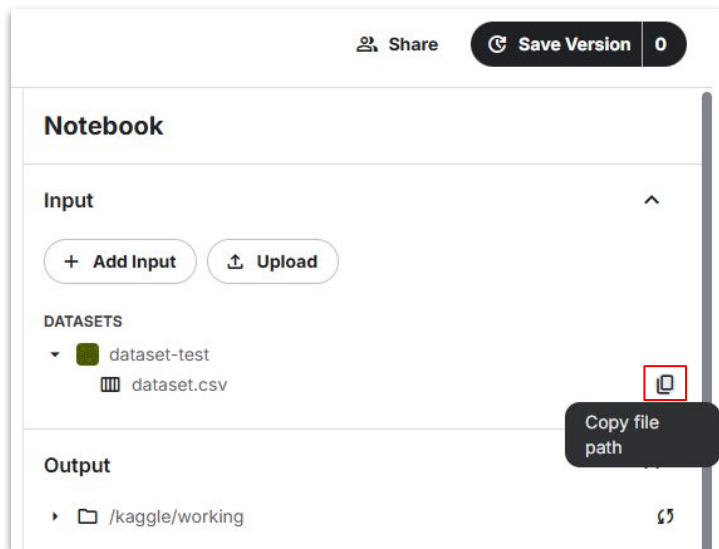
kaggle

- 上傳資料集 (方法二)



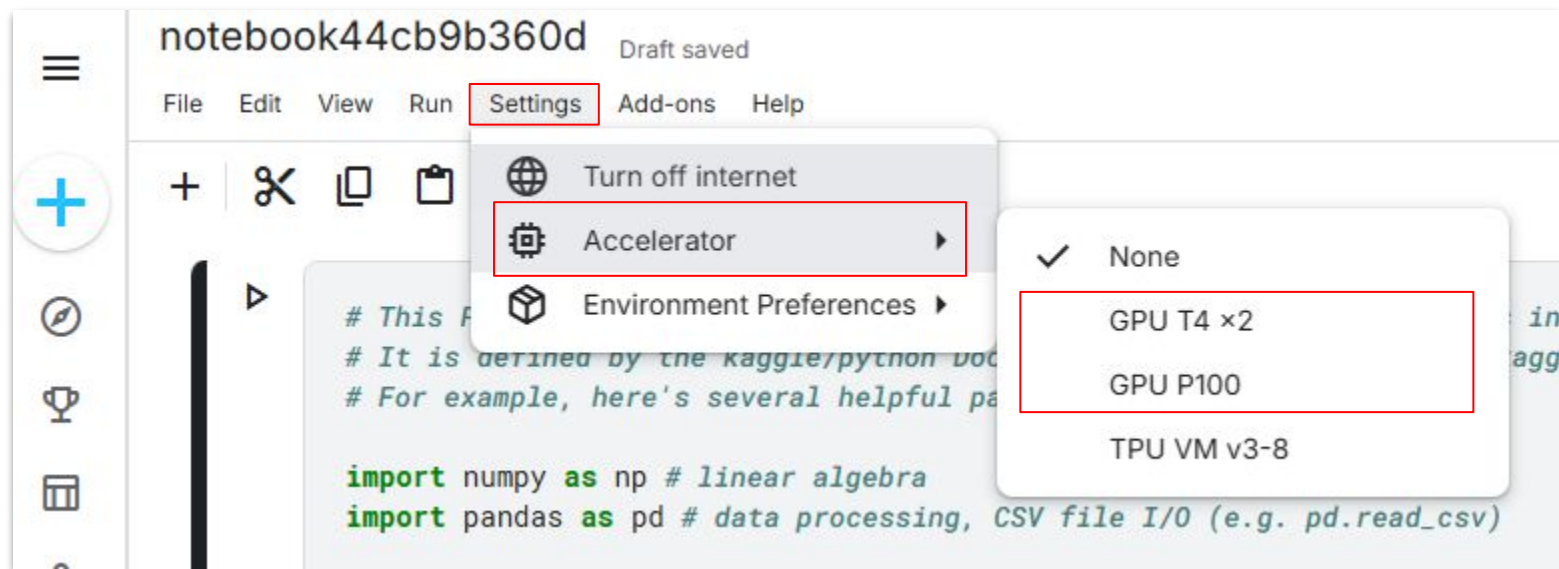
kaggle

- 引入資料集



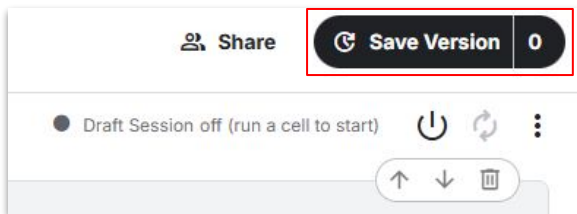
kaggle

- GPU設定



kaggle

- 儲存版本 (Quick Save)
 - 保存當下所有的程式及輸出 (不執行)



×

Save version

VERSION NAME

Version 1

9 / 50

VERSION TYPE

✓ Quick Save

Save a version of your notebook the way it currently looks

^ Advanced Settings

SAVE OUTPUT

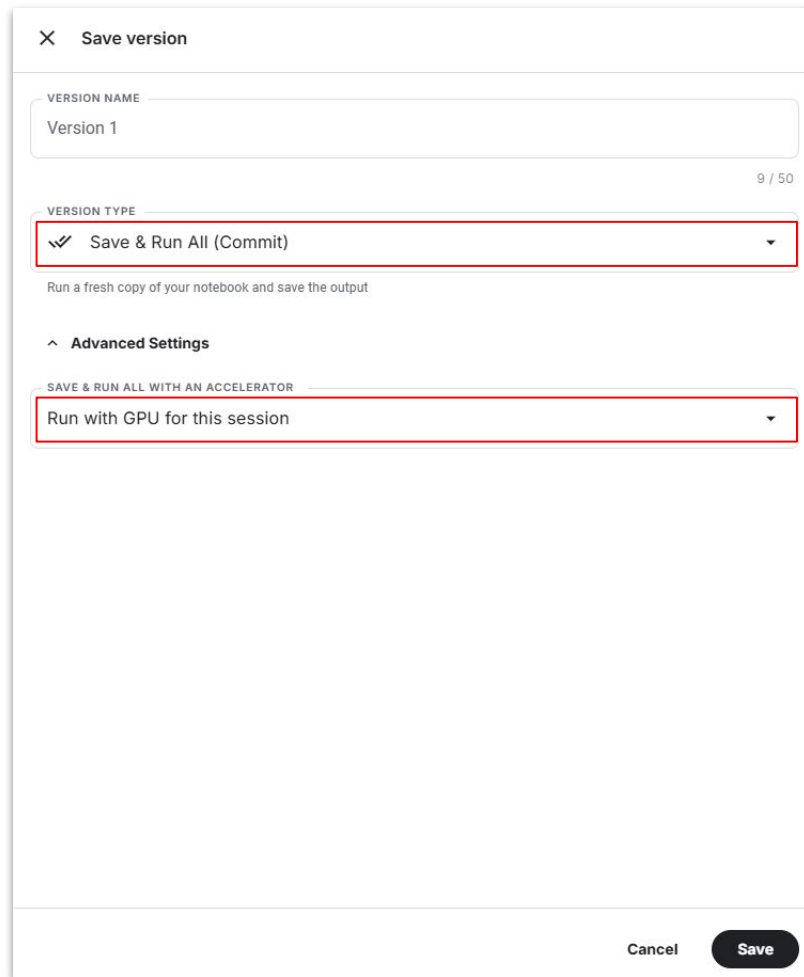
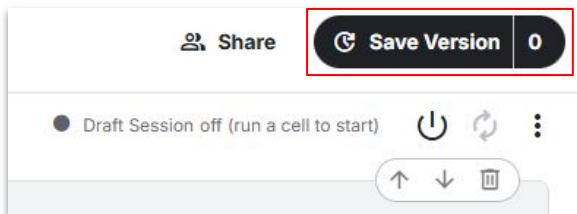
Always save output when creating a Quick Save

Cancel

Save


kaggle

- 儲存版本 (Run & Save)
 - 執行所有程式且保存輸出 (離線執行)



kaggle

- 使用 Invitation Link, 加入競賽



312511057 · COMMUNITY PREDICTION COMPETITION - PRIVATE - 3 DAYS TO GO

Join Competition

...

LLM hw

Overview

Data

Code

Models

Discussion

Leaderboard

Rules

Overview

News Classification

Start


3 minutes ago

Close

3 days to go

Competition Host

312511057



Prizes & Awards

Kudos

Does not award Points or Medals

Participation

0 Entrants

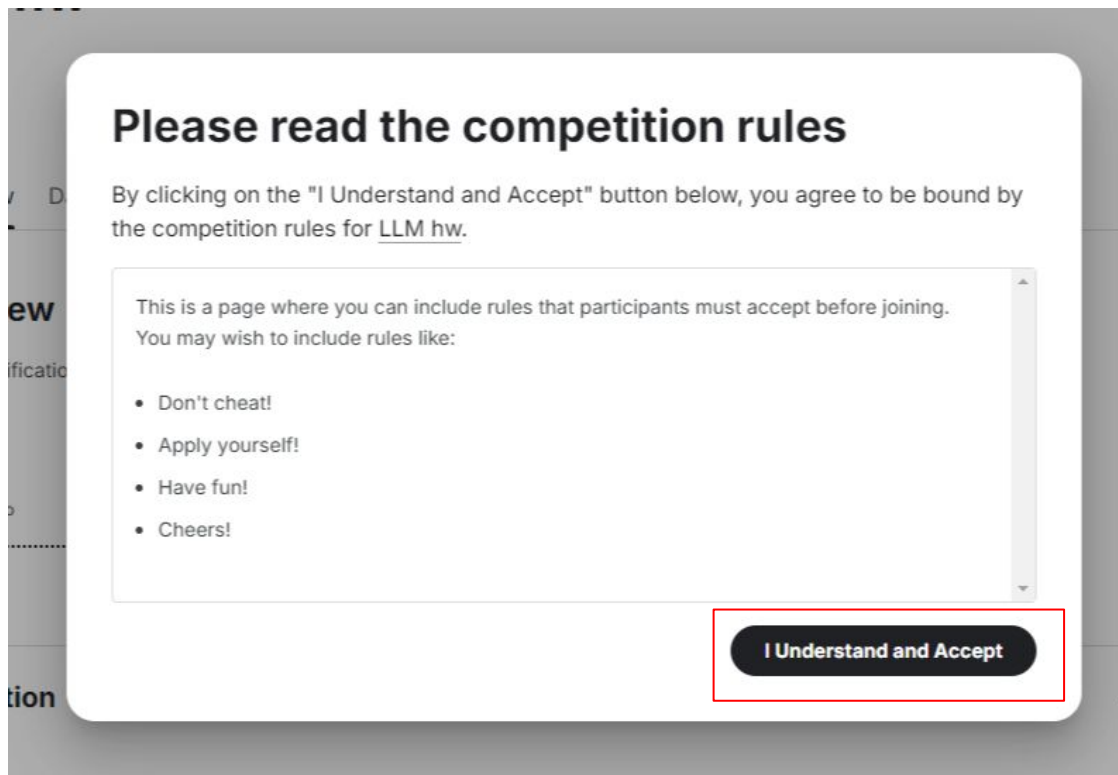
0 Participants

0 Teams

0 Submissions

kaggle

- 使用 Invitation Link, 加入競賽



Please read the competition rules

By clicking on the "I Understand and Accept" button below, you agree to be bound by the competition rules for LLM hw.

This is a page where you can include rules that participants must accept before joining.
You may wish to include rules like:

- Don't cheat!
- Apply yourself!
- Have fun!
- Cheers!

I Understand and Accept

The image shows a modal dialog box from Kaggle. It has a white background with rounded corners and a subtle drop shadow. The title "Please read the competition rules" is in bold black text. Below it is a paragraph explaining that clicking the "I Understand and Accept" button binds the user to the competition rules for "LLM hw". A text area follows, containing a placeholder message and a bulleted list of example rules. At the bottom right, there is a black button with white text "I Understand and Accept", which is highlighted by a red rectangular border.

kaggle

LLM hw

- Team Name

[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#)

Your Team

Everyone that competes in a Competition does so as a team - even if you're competing by yourself. [Learn more](#).

General

TEAM NAME

123456789

This name will appear on your team's leaderboard position.

Let others know you're looking for teammates

Your team can't accept more team members.

Team Members

Your team is at maximum capacity. Great job!

Save Changes

- Data Download

Dataset Description

File Description

train.jsonl - training set

test.jsonl - testing set for the leaderboard ranking and final scoring

sample_submission.csv - the sample submission file in the correct format

Data Fields

id - an anonymous id unique to a given news title

category - the label category of the news title

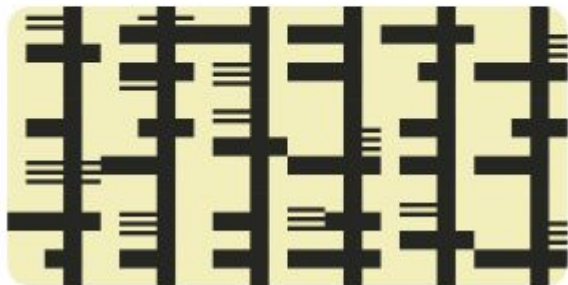
headline - the title of the news

short_description - news content

kaggle

- Submit Prediction

Submit Prediction



✕ Submit to Competition

File Upload Notebook



LLM hw

You have 5 submissions remaining today. This resets in 16 hours.



Drag and drop file to upload

(e.g., .csv, .parquet, .zip, .gz, .7z, .tar)

or

Browse Files

Your submission should be a CSV or Parquet file with 24,754 rows and a header. You can upload a zip/gz/7z/tar archive.

SUBMISSION DESCRIPTION

Enter a description

0 / 500

```
>_ kaggle competitions submit -c 6ab519d4fcd144a48ad4b3debe0ffd05
```



啟用 Windows
移至 [設定] 以啟用 Windows

Cancel

Submit

kaggle

- Leaderboard

Overview Data Code Models Discussion **Leaderboard** Rules Team Submissions


Leaderboard

 Raw Data

 Refresh

 Search leaderboard

This leaderboard is calculated with all of the test data.

#	Team	Members	Score	Entries	Last
	baseline		0.6521		

kaggle

常見問題

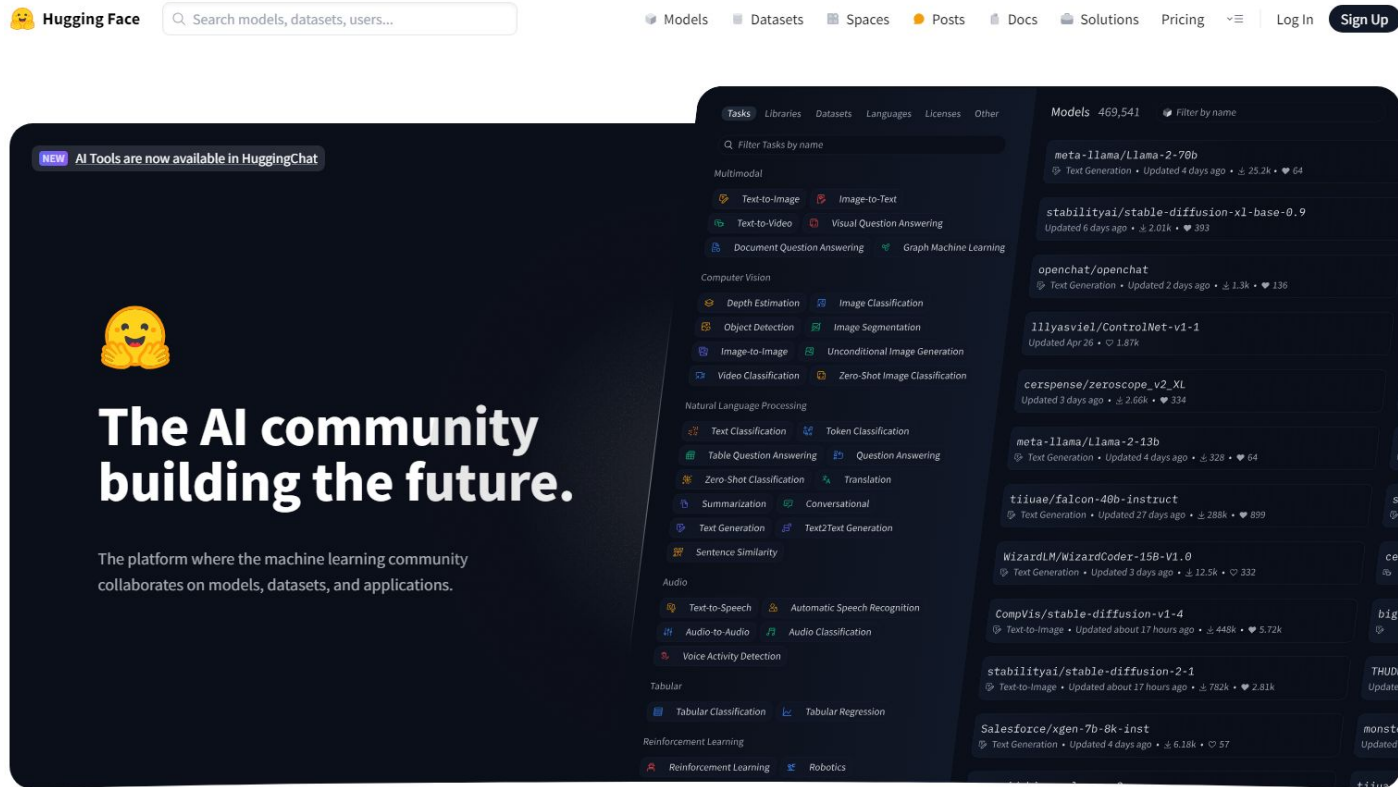
- 沒有GPU相關設定
 - 需註冊/登入帳號且綁定電話才能使用 GPU
- GPU使用時間限制
 - 一週30小時 (週六更新), 可至首頁右上角頭像 查詢
- 程式下載資源時出現錯誤
 - 上方Settings開啟網路功能



Hugging Face

Hugging Face

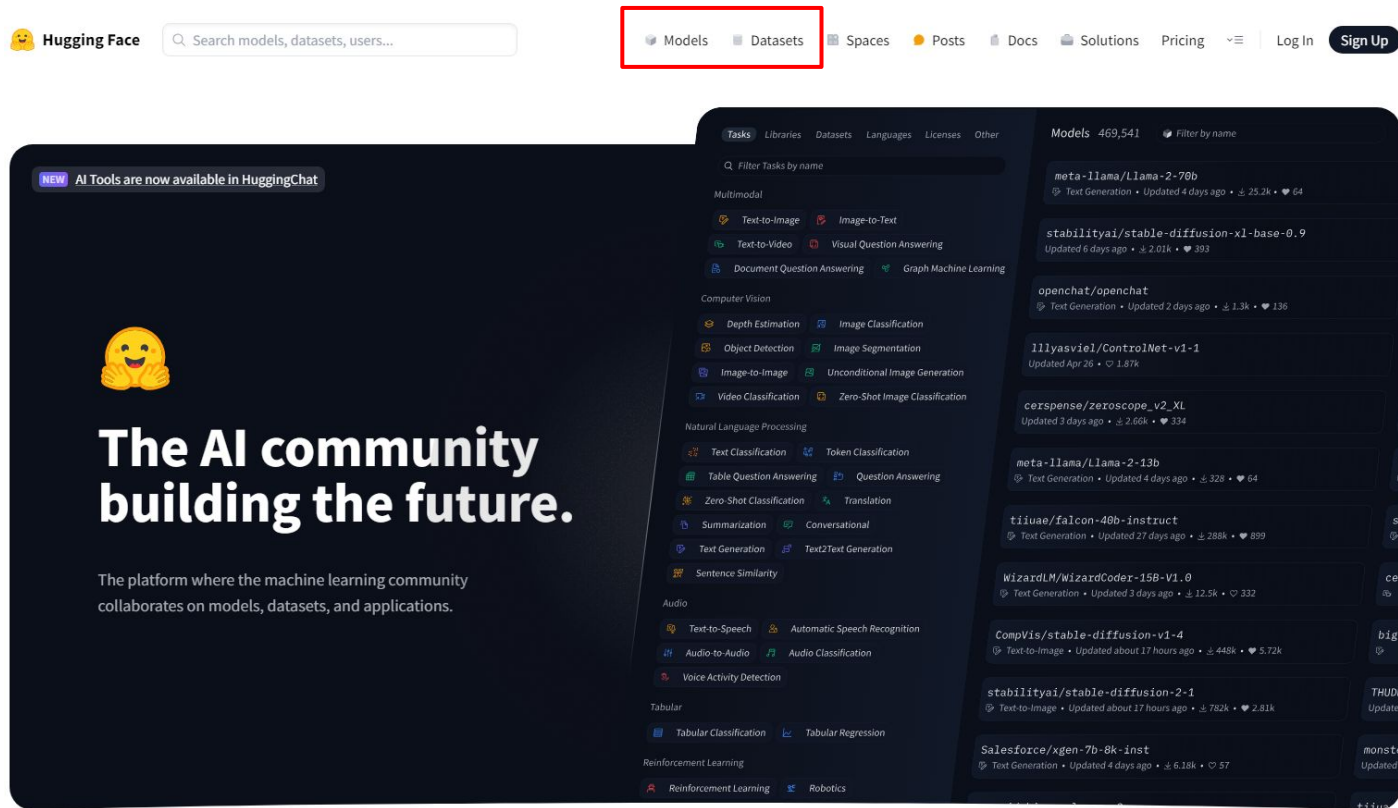
How to find the dataset/model



1-1 Switch to Hugging Face

Hugging Face

How to find the dataset/model



1-2 Click **Models** or **Datasets**

Hugging Face

How to find the dataset/model

The screenshot shows the Hugging Face homepage. At the top, there's a navigation bar with the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Posts, Docs, Solutions, Pricing, Log In, and Sign Up. Below the navigation bar, the main content area is divided into two sections. On the left, there's a sidebar with a red border containing the 'Tasks' section. This section is titled 'Filter Tasks by name' and lists various tasks under three categories: Multimodal, Computer Vision, and Natural Language Processing. On the right, there's a 'Models' section with a search bar and a 'Filter by name' button. Below this, there's a grid of model cards, each showing the model name, its capabilities, and some statistics like the number of likes and updates.

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering
- Video-Text-to-Text

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Models 867,513 Filter by name

Full-text search Sort: Trending

black-forest-labs/FLUX.1-dev
Text-to-Image • Updated 9 days ago • 479k • 3.14k

black-forest-labs/FLUX.1-schnell
Text-to-Image • Updated 9 days ago • 1.5M • 1.84k

microsoft/Phi-3.5-vision-instruct
Text Generation • Updated 1 day ago • 15.8k • 289

Mozilla/whisperfile
Updated 5 days ago • 1.73k • 174

city96/FLUX.1-dev-gguf
Text-to-Image • Updated 7 days ago • 101k • 293

XLabs-AI/flux-RealismLora
Text-to-Image • Updated 3 days ago • 142k • 455

nvidia/Mistral-NeMo-Minitron-8B-Base
Text Generation • Updated 3 days ago • 2.03k • 105

nvidia/Llama-3.1-Minitron-4B-Width-Base
Updated 3 days ago • 1.23k • 138

openbmb/MiniCPM-V-2_6
Image-Text-to-Text • Updated 3 days ago • 58.6k • 608

microsoft/Phi-3.5-MoE-instruct
Text Generation • Updated 5 days ago • 11k • 372

microsoft/Phi-3.5-mini-instruct
Text Generation • Updated 3 days ago • 50.1k • 319

XLabs-AI/flux-ip-adapter
Text-to-Image • Updated 3 days ago • 12.6k • 175

meta-llama/Meta-Llama-3.1-8B-Instruct
Text Generation • Updated 5 days ago • 2.65M • 2.02k

ai21labs/AI21-Jamba-1.5-Mini
Text Generation • Updated about 7 hours ago • 9.64k • 122

ai21labs/AI21-Jamba-1.5-Large
Text Generation • Updated about 7 hours ago • 107 • 105

akjindal53244/Llama-3.1-Storm-8B
Text Generation • Updated 5 days ago • 4.66k • 93

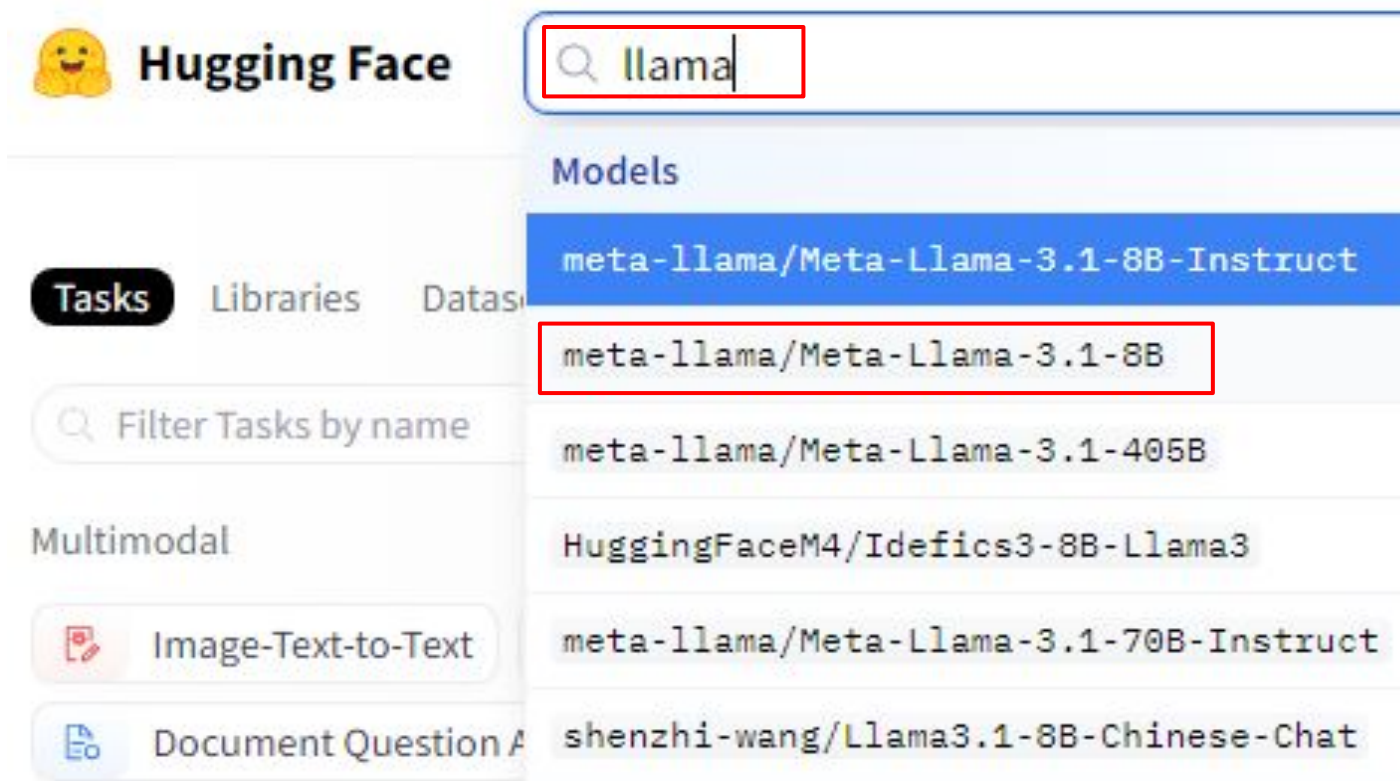
InstantX/FLUX.1-dev-Controlnet-Union
Updated 5 days ago • 4.87k • 221

lilyasviel/flux1-dev-bnb-nf4
Updated 11 days ago • 355

1-3 (Option) Click on the **task** option to filter your choices

Hugging Face

How to find the dataset/model



The screenshot shows the Hugging Face homepage. The search bar at the top right contains the text 'llama'. Below the search bar, a dropdown menu displays a list of models. The first model, 'meta-llama/Meta-Llama-3.1-8B-Instruct', is highlighted with a blue background. The second model, 'meta-llama/Meta-Llama-3.1-8B', is highlighted with a red border. The third model, 'meta-llama/Meta-Llama-3.1-405B', is also visible. The fourth model, 'HuggingFaceM4/Idefics3-8B-Llama3', is visible. The fifth model, 'meta-llama/Meta-Llama-3.1-70B-Instruct', is visible. The sixth model, 'shenzhi-wang/Llama3.1-8B-Chinese-Chat', is visible. On the left side of the page, there is a navigation bar with the Hugging Face logo and the text 'Hugging Face'. Below this, there are tabs for 'Tasks', 'Libraries', and 'Datasets'. The 'Tasks' tab is selected. Below the tabs, there is a search bar with the text 'Filter Tasks by name'. Below the search bar, there is a section titled 'Multimodal'. Under this section, there are two buttons: 'Image-Text-to-Text' and 'Document Question Answering'.

Hugging Face

Tasks Libraries Datasets

Filter Tasks by name

Multimodal

Image-Text-to-Text

Document Question Answering

Models

meta-llama/Meta-Llama-3.1-8B-Instruct

meta-llama/Meta-Llama-3.1-8B

meta-llama/Meta-Llama-3.1-405B

HuggingFaceM4/Idefics3-8B-Llama3

meta-llama/Meta-Llama-3.1-70B-Instruct

shenzhi-wang/Llama3.1-8B-Chinese-Chat

1-4 Enter your **model/dataset name** and click on the model you want to select

Hugging Face

How to find the dataset/model

The screenshot shows the Hugging Face interface for the model **meta-llama/Meta-Llama-3.1-8B**. The model has 627 likes. Below the title are tags for Text Generation, Transformers, Safetensors, PyTorch, 8 languages, llama, facebook, meta, and llama-3. The navigation bar includes Model card (selected), Files and versions, and Community (36). An 'Edit model card' link is visible. A modal dialog is open with the title 'You need to agree to share your contact information to access this model'. The dialog contains text about data collection and sharing, followed by the 'LLAMA 3.1 COMMUNITY LICENSE AGREEMENT' section. This section includes the release date (July 23, 2024), definitions for 'Agreement', 'Documentation', and 'Licensee', and buttons for 'Log in' or 'Sign Up' to review conditions and access the model content.

meta-llama / **Meta-Llama-3.1-8B** like 627

Text Generation Transformers Safetensors PyTorch 8 languages llama facebook meta llama-3

Model card Files and versions Community 36

Edit model card

You need to agree to share your contact information to access this model

The information you provide will be collected, stored, processed and shared in accordance with the [Meta Privacy Policy](#).

LLAMA 3.1 COMMUNITY LICENSE AGREEMENT

Llama 3.1 Version Release Date: July 23, 2024

"Agreement" means the terms and conditions for use, reproduction, distribution and modification of the Llama Materials set forth herein.

"Documentation" means the specifications, manuals and documentation accompanying Llama 3.1 distributed by Meta at <https://llama.meta.com/doc/overview>.




"Licensee" or "you" means you, or your employer or any other person or entity (if you are entering into this Agreement on such person or...










Log in or Sign Up to review the conditions and access this model content.




1-5 You **successfully** found the model/dataset you wanted!





Hugging Face


How to use the model


 meta-llama / **Meta-Llama-3.1-8B**   like 627

 Text Generation  Transformers  Safetensors  PyTorch  8 languages llama facebook meta llama-3  text-generation-inference  Inference Endpoints  arxiv:2204.05149  License: llama3.1

 **Model card**  Files and versions  Community **36**

  Train  Deploy  Use this model

 Edit model card

 **You need to agree to share your contact information to access this model**

The information you provide will be collected, stored, processed and shared in accordance with the [Meta Privacy Policy](#).



LLAMA 3.1 COMMUNITY LICENSE AGREEMENT

Llama 3.1 Version Release Date: July 23, 2024



"Agreement" means the terms and conditions for use, reproduction, distribution and modification of the Llama Materials set forth herein.


"Documentation" means the specifications, manuals and documentation accompanying Llama 3.1 distributed by Meta at <https://llama.meta.com/doc/overview>.



"Licensee" or "you" means you, or your employer or any other person or entity (if you are entering into this Agreement on such person or...



 Log in or  Sign Up to review the conditions and access this model content.




Downloads last month
473,558

 **Safetensors** 

Model size 8.03B params Tensor type BF16 

 **Inference API** 

 Cold 

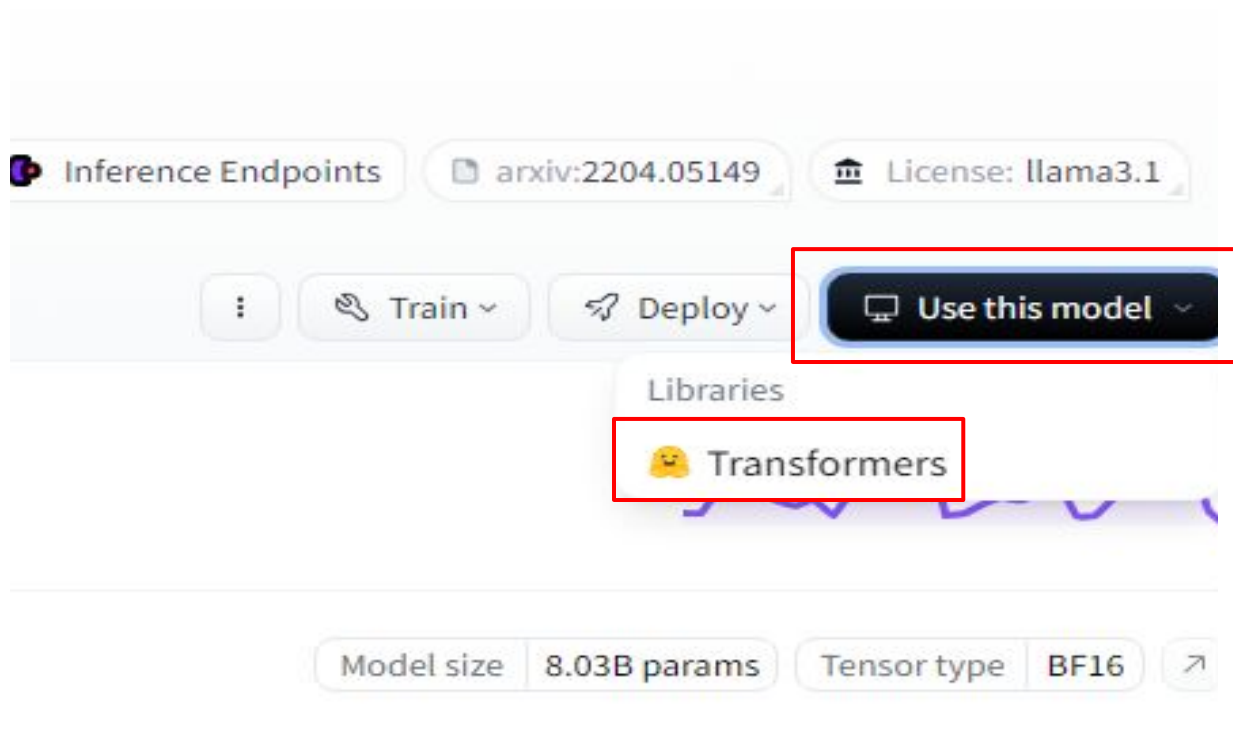
 Text Generation  Examples 

My name is Merve and my favorite

2-1 Switch to the [model page](#)

Hugging Face

How to use the model



2-2 Click **Use this model** and click the libraries

Hugging Face

How to use the model

How to use from the **Transformers** library



Use a pipeline as a high-level helper

Copy

```
from transformers import pipeline
```

```
pipe = pipeline("text-generation", model="meta-llama/Meta-Llama-3.1-8B")
```

Load model directly

Copy

```
from transformers import AutoTokenizer, AutoModelForCausalLM
```

```
tokenizer = AutoTokenizer.from_pretrained("meta-llama/Meta-Llama-3.1-8B")
```

```
model = AutoModelForCausalLM.from_pretrained("meta-llama/Meta-Llama-3.1-8B")
```

Quick Links

- 🔗 Read model documentation
- 🔗 Read docs on high-level-pipeline
- 🔗 Read our learning resources

2-3 You can copy the code below the words "Load model directly" to use it!

Hugging Face

Reference link

- <https://huggingface.co/>
- 官方教程
 - <https://huggingface.co/learn/nlp-course/chapter1/1>
 - <https://www.youtube.com/watch?v=QEaBAZQCtwE>

Natural Language Processing

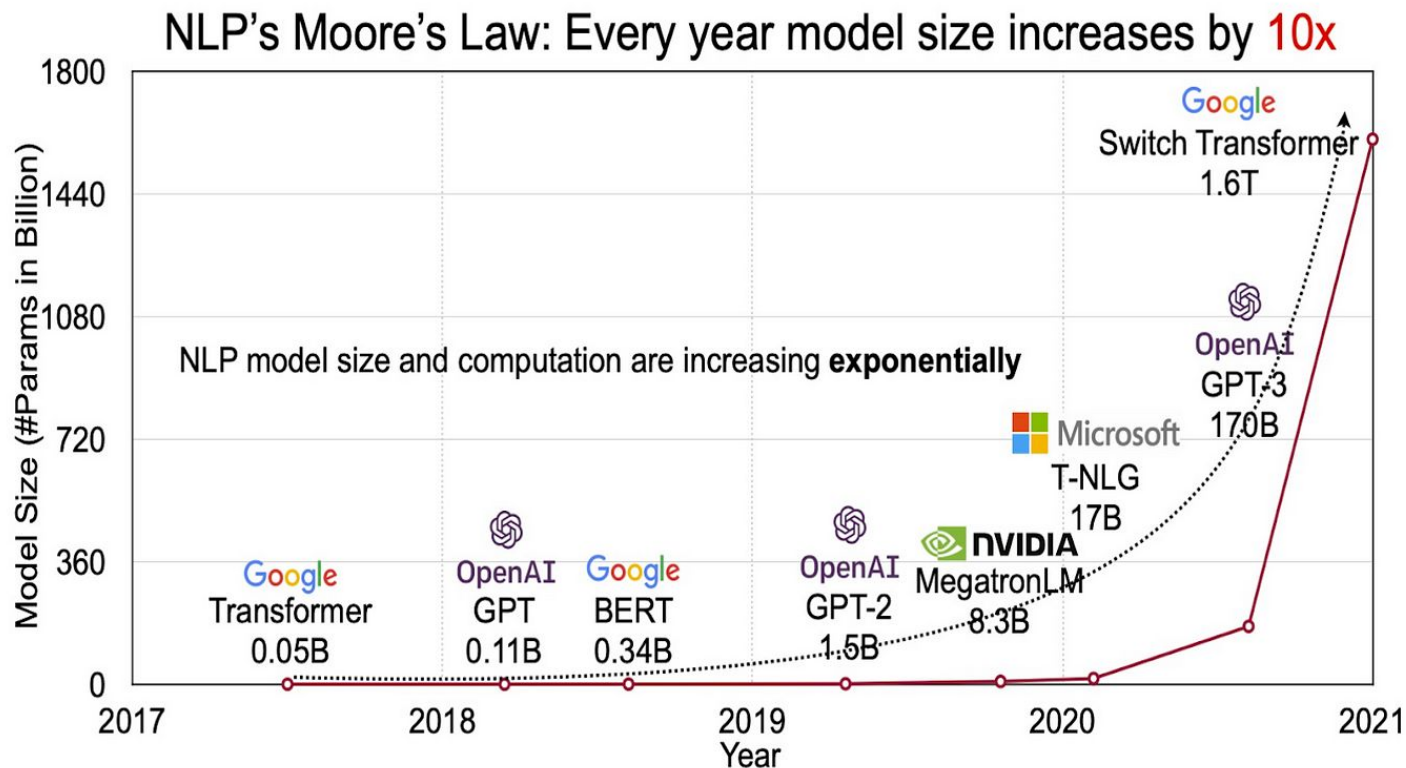
What is NLP ?

- Classifying each word in a sentence
- Classifying whole sentences
- Extracting an answer from a text
- Generating a new sentence from an input text
- Generating text content

LM History

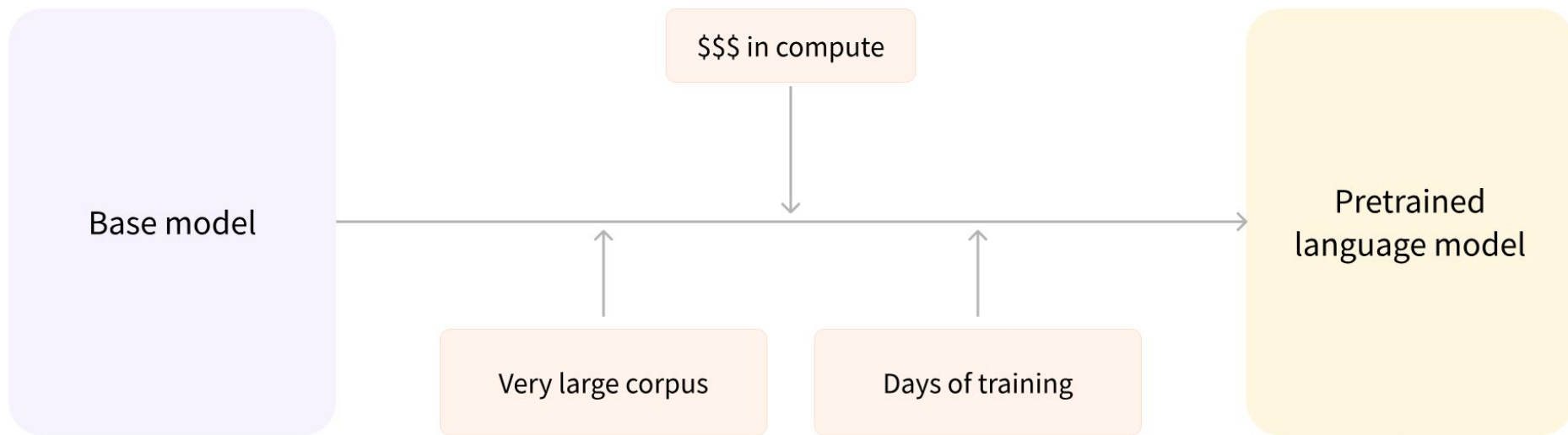


LM Size



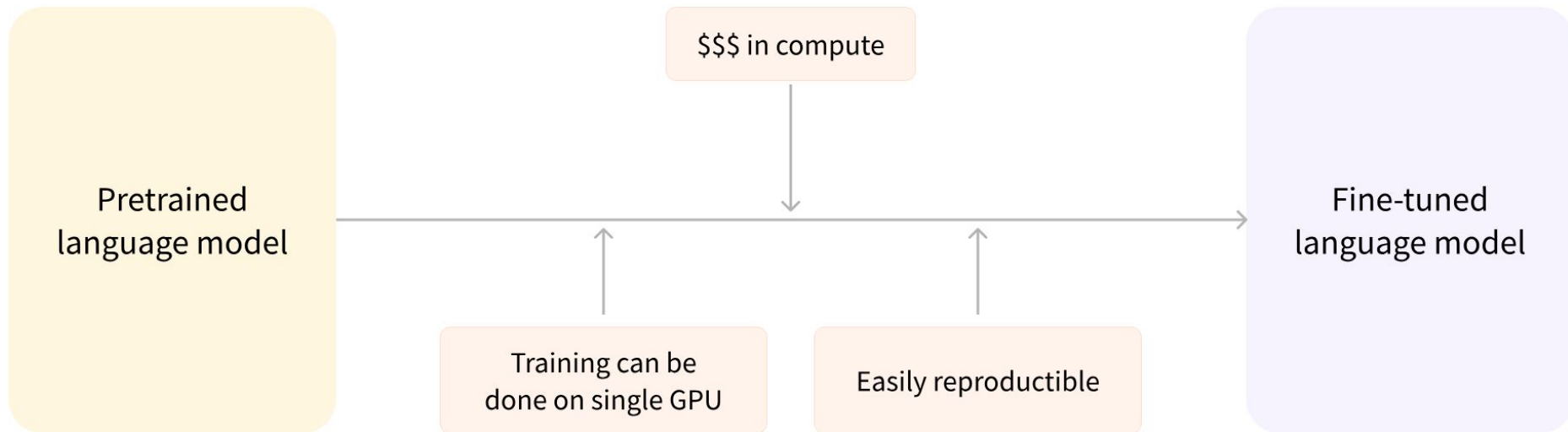
Transfer Learning

- Pretraining : training a model from scratch

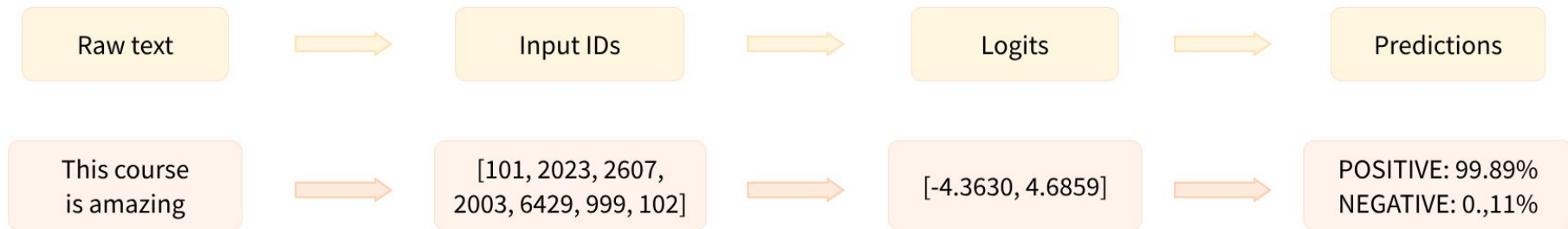
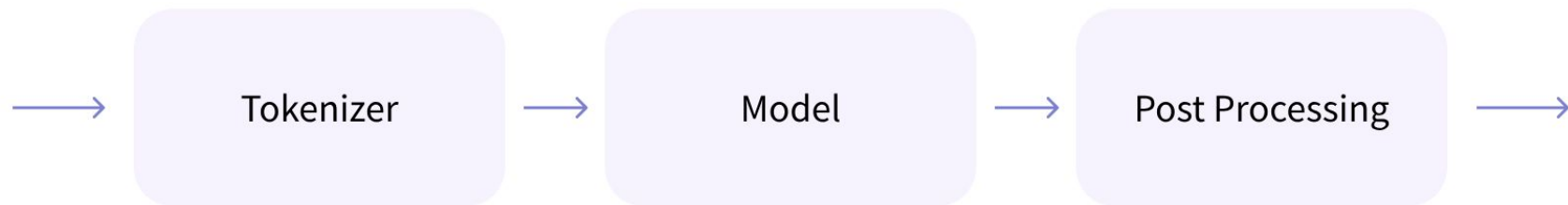


Transfer Learning

- Fine-tuning : the training done after a model has been pretrained



NLP Task (Sentiment Analysis)



Tokenizer

- Splitting the input into words, subwords, or symbols (like punctuation) that are called tokens
- Mapping each token to an integer
- Adding additional inputs that may be useful to the model

Sample Data:

"This is tokenizing."

Character Level

[T] [h] [i] [s] [i] [s] [t] [o] [k] [e] [n] [i] [z] [i] [n] [g] [.]

Word Level

[This] [is] [tokenizing] [.]

Subword Level

[This] [is] [token] [izing] [.]

Tokenizer

```
from transformers import AutoTokenizer

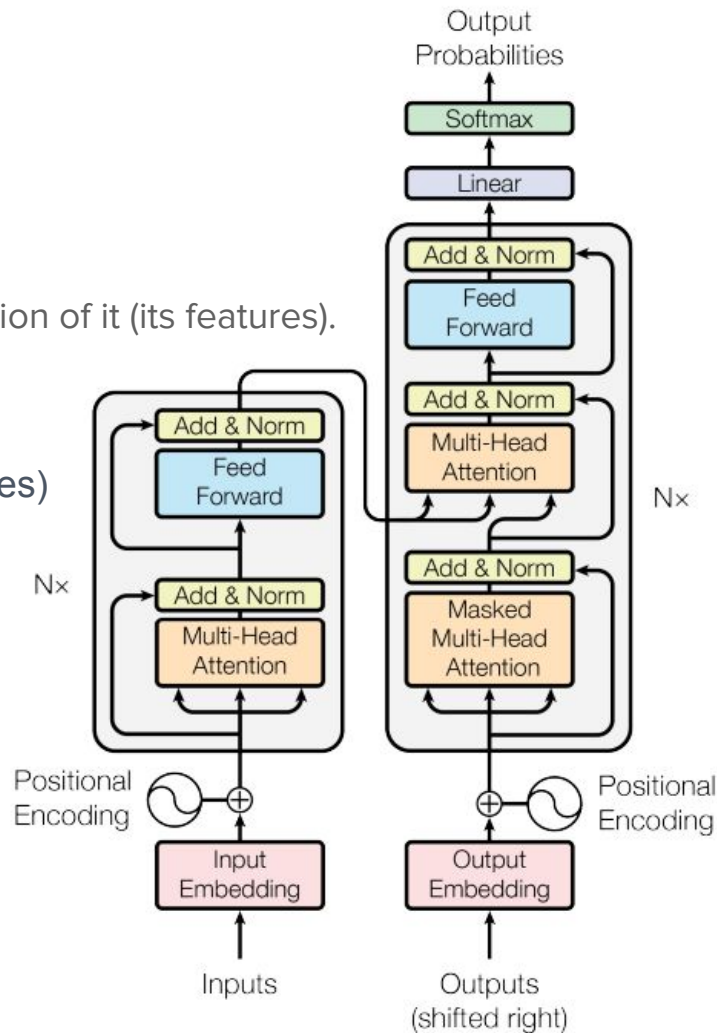
checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)

raw_inputs = [
    "I've been waiting for a HuggingFace course my whole life.",
    "I hate this so much!",
]
inputs = tokenizer(raw_inputs, padding=True, truncation=True, return_tensors="pt")
print(inputs)
```

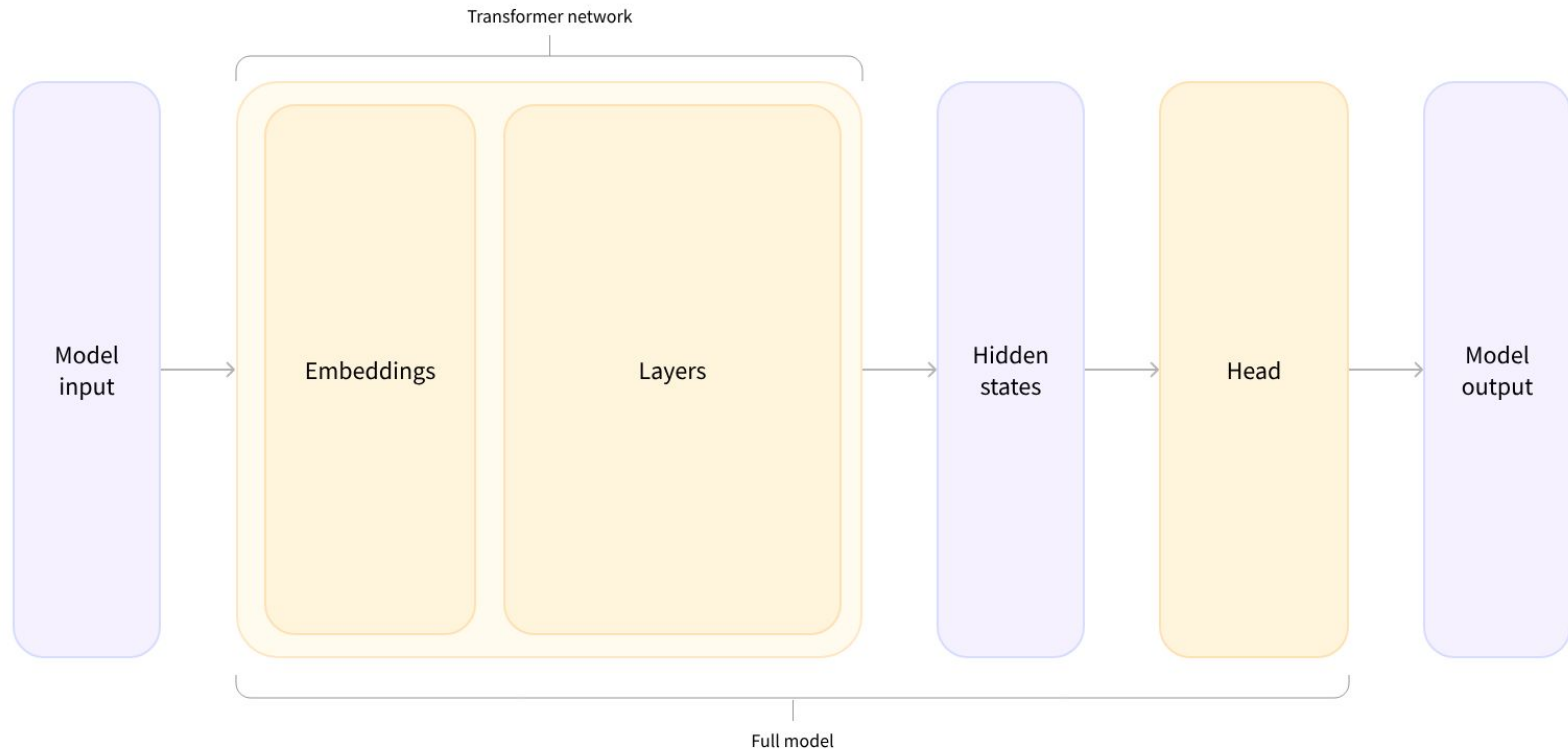
```
{
  'input_ids': tensor([
    [ 101, 1045, 1005, 2310, 2042, 3403, 2005, 1037, 17662, 12172, 2607, 2026, 2878, 2166, 1012, 102],
    [ 101, 1045, 5223, 2023, 2061, 2172, 999, 102, 0, 0, 0, 0, 0, 0, 0, 0]
  ]),
  'attention_mask': tensor([
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]
  ])
}
```

Transformer

- Encoder
 - The encoder receives an input and builds a representation of it (its features).
 - BERT, RoBERTa
- Decoder
 - The decoder uses the encoder's representation (features) along with other inputs to generate a target sequence.
 - GPT, GPT-2



Model



Model

```
from transformers import AutoModelForSequenceClassification

checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
model = AutoModelForSequenceClassification.from_pretrained(checkpoint)
outputs = model(**inputs)
```

```
print(outputs.logits)
```

```
tensor([[ -1.5607,  1.6123],
        [ 4.1692, -3.3464]], grad_fn=<AddmmBackward>)
```


Postprocessing the Output

- "I've been waiting for a HuggingFace course my whole life."
 - NEGATIVE: 0.0402, POSITIVE: 0.9598
- "I hate this so much!"
 - NEGATIVE: 0.9995, POSITIVE: 0.0005

```
import torch

predictions = torch.nn.functional.softmax(outputs.logits, dim=-1)
print(predictions)
```

```
tensor([[4.0195e-02, 9.5980e-01],
        [9.9946e-01, 5.4418e-04]], grad_fn=<SoftmaxBackward>)
```

Fine-tuning a model with the Trainer API

- Loading a dataset

```
from datasets import load_dataset
from transformers import AutoTokenizer, DataCollatorWithPadding

raw_datasets = load_dataset("glue", "mrpc")
checkpoint = "bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)

def tokenize_function(example):
    return tokenizer(example["sentence1"], example["sentence2"], truncation=True)

tokenized_datasets = raw_datasets.map(tokenize_function, batched=True)
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

Fine-tuning a model with the Trainer API

- Training

```
from transformers import TrainingArguments
```

```
training_args = TrainingArguments("test-trainer")
```

```
from transformers import AutoModelForSequenceClassification
```

```
model = AutoModelForSequenceClassification.from_pretrained(checkpoint, num_labels=2)
```

```
from transformers import Trainer
```

```
trainer = Trainer(  
    model,  
    training_args,  
    train_dataset=tokenized_datasets["train"],  
    eval_dataset=tokenized_datasets["validation"],  
    data_collator=data_collator,  
    tokenizer=tokenizer,  
)
```

```
trainer.train()
```

Fine-tuning a model with the Trainer API

- Evaluation

```
predictions = trainer.predict(tokenized_datasets["validation"])  
print(predictions.predictions.shape, predictions.label_ids.shape)
```

```
import numpy as np  
  
preds = np.argmax(predictions.predictions, axis=-1)
```

```
import evaluate  
  
metric = evaluate.load("glue", "mrpc")  
metric.compute(predictions=preds, references=predictions.label_ids)
```

```
{'accuracy': 0.8578431372549019, 'f1': 0.8996539792387542}
```

RAG and LangChain

檢索增強生成(RAG)

目的：解決特定外部知識問題



你

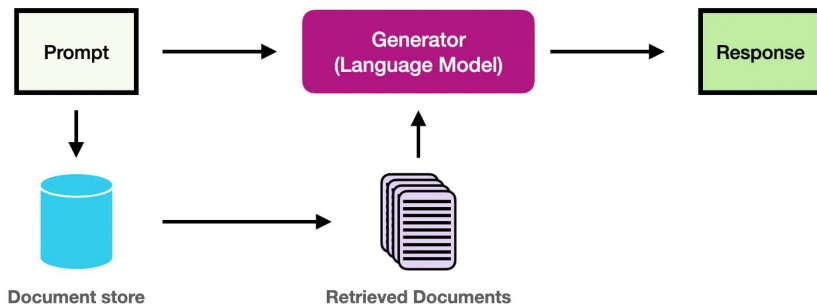
請你介紹今年4/3發生在台灣的大地震



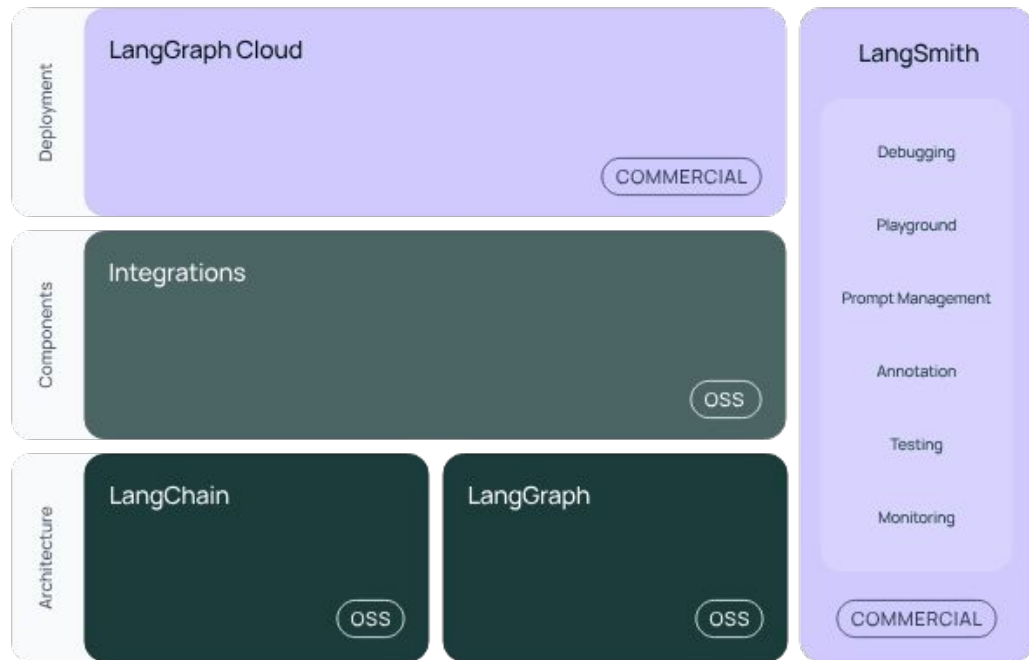
ChatGPT

抱歉，我無法提供關於最近的自然災害或其他事件的即時資訊。你可以通過查閱新聞來獲取有關任何事件的最新信息，或者你有其他想知道的事情嗎？我很樂意幫助你。

Naïve RAG

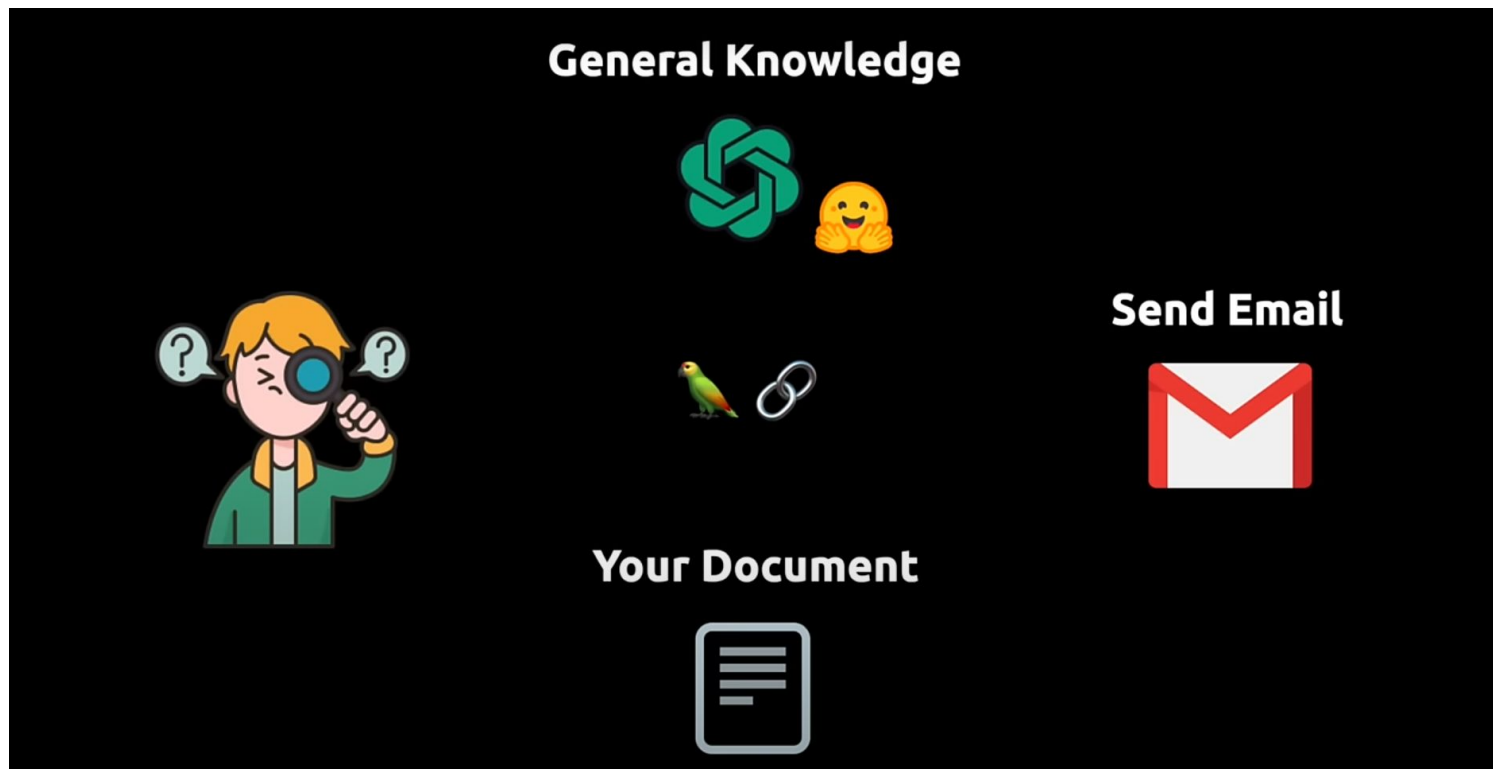


Langchain v0.2

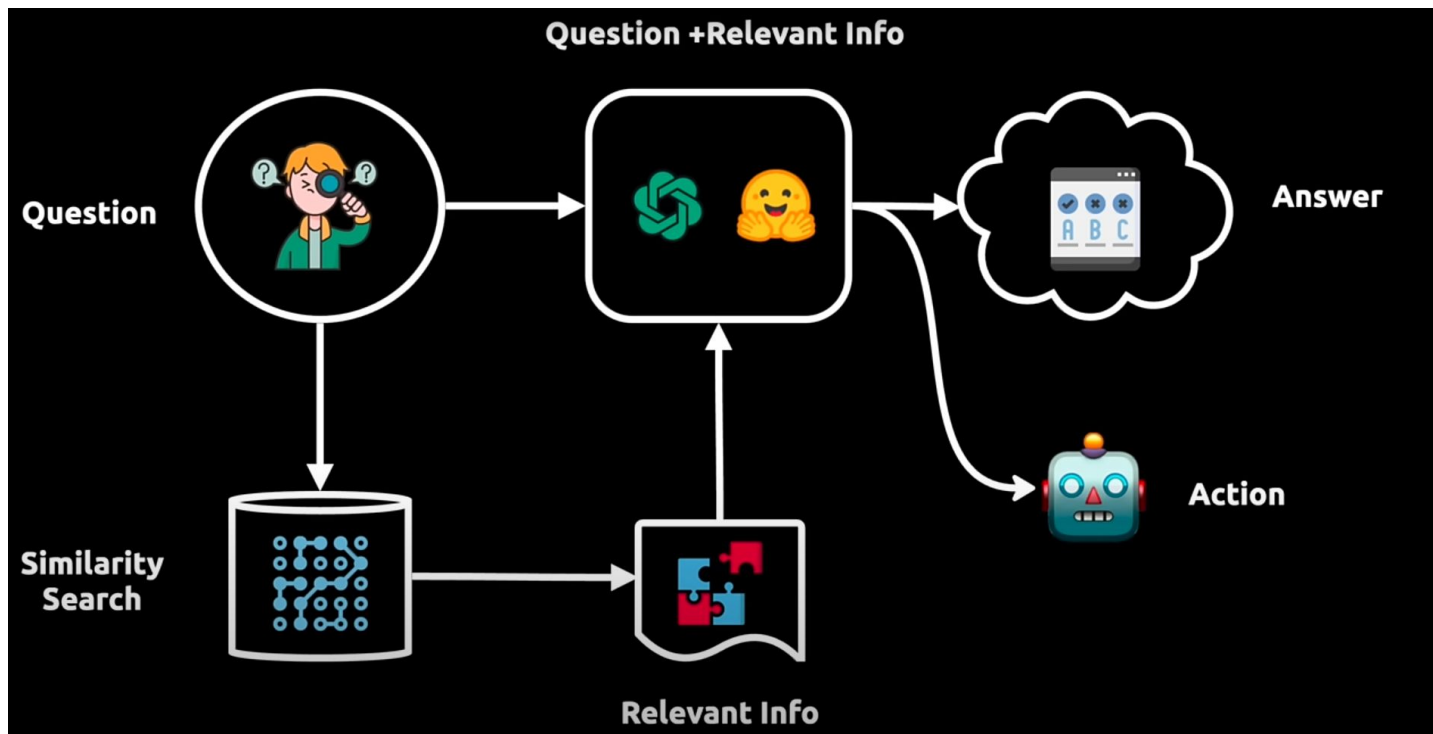


- **開發**: 使用 LangChain 的開源元件、LangGraph 和第三方的 package 來構建應用程式。
- **生產化**: 使用 LangSmith 檢查、監控和評估 langchain 程式，以便持續優化並部署。
- **部署**: 通過 LangGraph Cloud 將 LangGraph 應用程式轉換為 API。

Why Langchain?



Why Langchain?



LangChain main concepts

Components



- LLM Wrappers
- Prompt Templates
- Indexes for relevant information retrieval

Chains



Assemble components to solve a specific task, such as finding info in a book.

Agents



Agents allow LLMs to interact with its environment. For instance, make API request with a specific action.

RAG two main concepts

1. **Indexing**: a pipeline for ingesting data from a source and indexing it. This usually happens offline.
2. **Retrieval and generation**: the actual RAG chain, which takes the user query at run time and retrieves the relevant data from the index, then passes that to the model.

Indexing: Load

```
import bs4
from langchain_community.document_loaders import WebBaseLoader

# Only keep post title, headers, and content from the full HTML.
bs4_strainer = bs4.SoupStrainer(class_=("post-title", "post-header", "post-content"))
loader = WebBaseLoader(
    web_paths=("https://lilianweng.github.io/posts/2023-06-23-agent/",),
    bs_kwargs={"parse_only": bs4_strainer},
)
docs = loader.load()

len(docs[0].page_content)
```



Indexing: Load

```
print(docs[0].page_content[:500])
```



LLM Powered Autonomous Agents

Date: June 23, 2023 | Estimated Reading Time: 31 min | Author: Lilian Weng

Building agents with LLM (large language model) as its core controller is a cool concept. Several proof.
Agent System Overview#
In

Indexing: Split

```
from langchain_text_splitters import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=1000, chunk_overlap=200, add_start_index=True
)
all_splits = text_splitter.split_documents(docs)

len(all_splits)
```



Indexing: Split

66

```
len(all_splits[0].page_content)
```

969

```
all_splits[10].metadata
```

```
{'source': 'https://lilianweng.github.io/posts/2023-06-23-agent/',  
 'start_index': 7056}
```

Indexing: Store

```
from langchain_chroma import Chroma
from langchain_openai import OpenAIEmbeddings

vectorstore = Chroma.from_documents(documents=all_splits, embedding=OpenAIEmbeddings())
```


Retrieval and Generation: Retrieve

```
retriever = vectorstore.as_retriever(search_type="similarity", search_kwargs={"k": 6})  
  
retrieved_docs = retriever.invoke("What are the approaches to Task Decomposition?")  
  
len(retrieved_docs)
```

6

```
print(retrieved_docs[0].page_content)
```

Tree of Thoughts (Yao et al. 2023) extends CoT by exploring multiple reasoning possibilities at each step. Task decomposition can be done (1) by LLM with simple prompting like "Steps for XYZ.\n1.", "What are the

Retrieval and Generation: Generate

```
import getpass
import os

os.environ["OPENAI_API_KEY"] = getpass.getpass()

from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model="gpt-4o-mini")
```



Retrieval and Generation: Generate

```
from langchain import hub

prompt = hub.pull("rlm/rag-prompt")

example_messages = prompt.invoke(
    {"context": "filler context", "question": "filler question"}
).to_messages()

example_messages
```

```
[HumanMessage(content="You are an assistant for question-answering tasks. Use the following pieces of re
```

```
print(example_messages[0].content)
```

```
You are an assistant for question-answering tasks. Use the following pieces of retrieved context to ans
Question: filler question
Context: filler context
Answer:
```

Retrieval and Generation: Generate

```
def format_docs(docs):  
    return "\n\n".join(doc.page_content for doc in docs)  
  
rag_chain = (  
    {"context": retriever | format_docs, "question": RunnablePassthrough()}  
    | prompt  
    | llm  
    | StrOutputParser()  
)  
  
rag_chain.invoke("What is Task Decomposition?")
```

'Task Decomposition is a process where a complex task is broken down into smaller, simpler steps or sub

加簽

- 會從表單中抽十位同學加簽
- 注意加簽的同學期中不能退選，請同學慎重考慮

