NKUST

生成式AI設計與實作

Presentation：沈世賢

OUTLINE

01 生成式AI介紹

02 Ollama 介紹

03 提升模型表現方式

04 Open webUI

05 RAG 模型實作

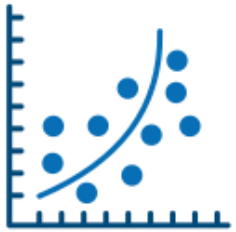
01 生成式人工智慧(Generative AI)

一般AI

分類

偵測

回歸



生成式AI

讓機器產生複雜且有結構的物件

文章

圖片

語音

01 生成式人工智慧(Generative AI)

人工智慧(目標)

機器學習(手段)

深度學習(更厲害的手段)

生成式人工智慧(目標之一)

生成式人工智慧現今大多依靠深度學習來實現

01 如何產生訓練時沒看過的東西?

訓練資料

輸入: 何謂人工智慧？ 輸出: 人工智慧就是 .....

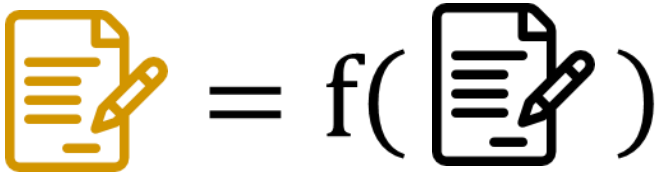
輸入: 說個跟人工智慧有關的故事 輸出: 很久很久以前 .....

輸入: 寫一首詩 輸出: 床前明月光 .....

輸入: 人工智慧的英文翻譯 輸出: Artificial Intelligence (AI)

測試模型

需創造全新的資料



文字：寫一篇題為「縫隙的聯想」的文章

01

核心方法 文字接龍

原本的目標　台灣的最高山是哪座山？ → 函式 → 玉山

可能性
無窮無盡

拆解成一連串文字接龍　分類問題

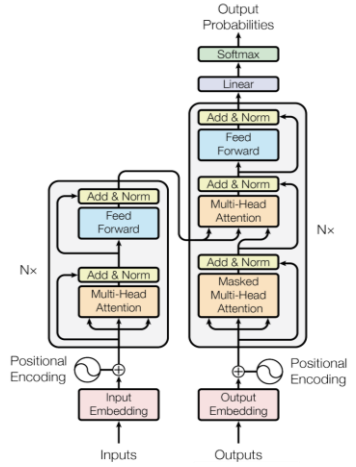
答案有限!!

台灣的最高山是哪座山？ → 函式 → 玉

台灣的最高山是哪座山？ 玉 → 函式 → 山

台灣的最高山是哪座山？ 玉山 → 函式 → [END]

語言模型



Transformer

02

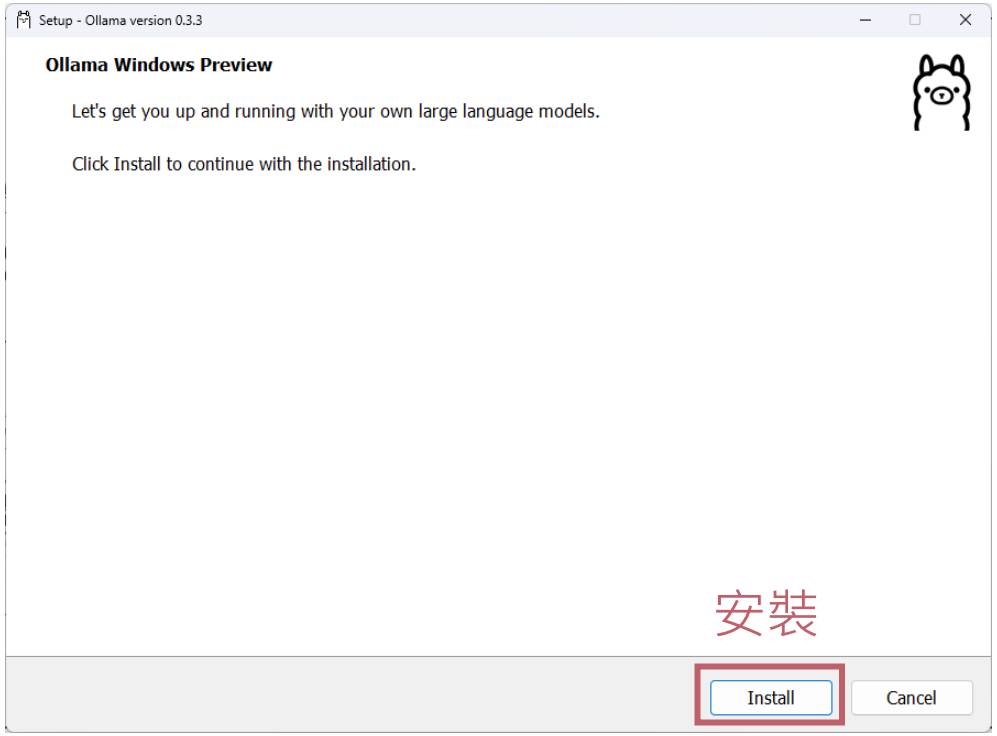
Ollama 介紹

Ollama 是一個開源軟體，讓使用者可以在自己的硬體上運行、創建和分享大型語言模型服務。這個平台適合希望在本地端運行模型的使用者，因為它不僅可以保護隱私，還允許用戶透過命令行介面輕鬆地設置和互動。Ollama 支援包括 Llama 3 和 Mistral 等多種模型，並提供彈性的客製化選項，例如從其他格式導入模型並設置運行參數。

02

Ollama 安裝

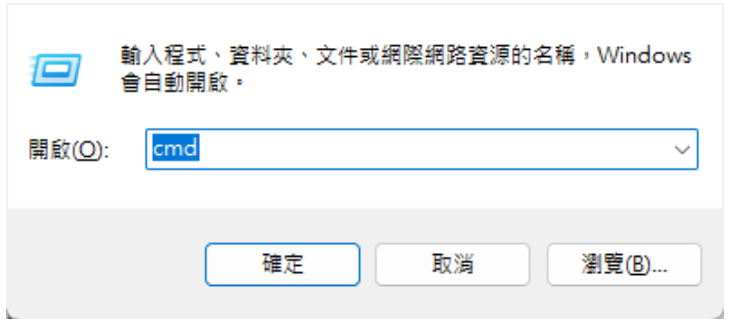
https://ollama.com/



02

Ollama 指令

1. Win + R 輸入cmd 開啟命令提示窗口



2. 輸入 ollama –v 查看軟體版本

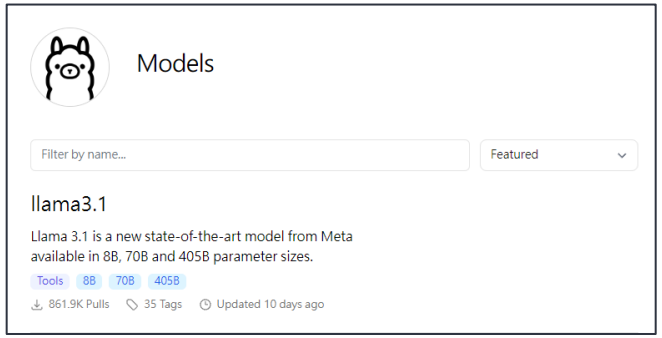
```
❯  ollama -v
ollama version is 0.3.0
```

3. 輸入 ollama –h 查看幫助

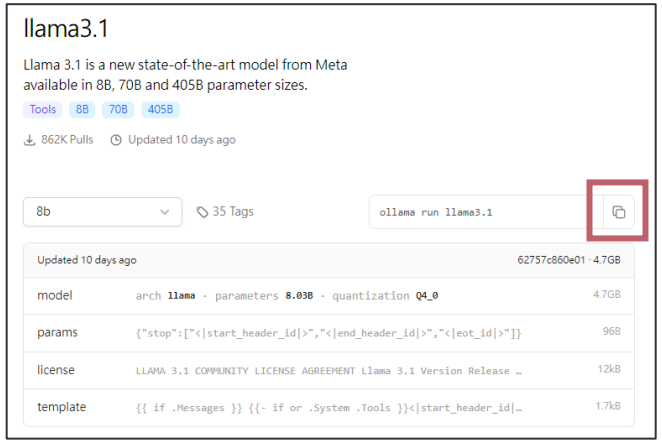
```
❯  ollama -h
Large language model runner

Usage:
  ollama [flags]
  ollama [command]
```

4. 尋找可用模型,以llama3.1為例

https://ollama.com/library



5. 點選複製按鈕，複製運行指令



6. 將複製的文字貼入cmd中即可開始對話囉

```
❯  ollama run llama3.1
>>> Send a message (/? for help)
```

```
>>> 台灣最高的山是哪座山
玉山
```

7. 輸入 /bye 即可退出對話

8. 在退出後輸入 ollama list 查看本地模型

```
❯  ollama list
NAME                    ID              SIZE    MODIFIED
llama3.1:latest         62757c860e01    4.7 GB  6 days ago
llama3_TAIDE:latest     b4d373d5cd4c    4.9 GB  4 weeks ago
llama3:latest           365c0bd3c000    4.7 GB  4 weeks ago
```

9. 輸入 ollama rm <model> 來刪除模型

```
❯  ollama rm llama3_TAIDE
deleted 'llama3_TAIDE'
```

```
❯  ollama list
NAME            ID              SIZE    MODIFIED
llama3.1:latest 62757c860e01    4.7 GB  6 days ago
llama3:latest   365c0bd3c000    4.7 GB  4 weeks ago
```

03

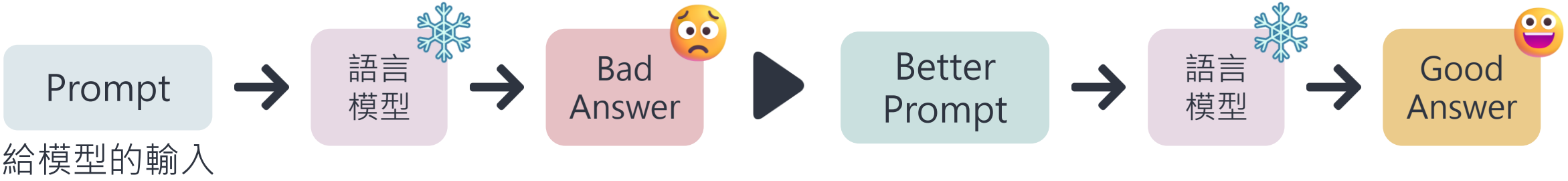
提升模型表現方式

思路一：我改變不了模型，那我改變我自己！

與其調整函式，不如調整人

給更清楚的指令，提供額外資訊

Prompt Engineering 提示詞工程

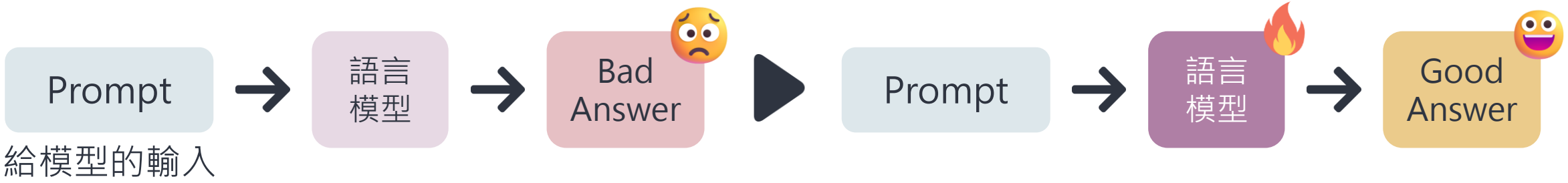


03

提升模型表現方式

思路二：我要自己訓練一個模型！

Fine Tuning 微調模型

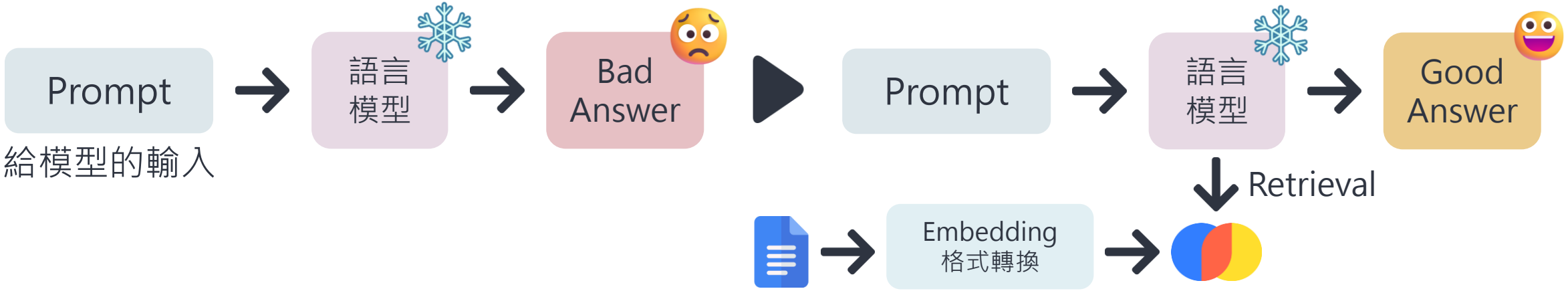


03

提升模型表現方式

思路三：我要讓模型先查資料在回答！

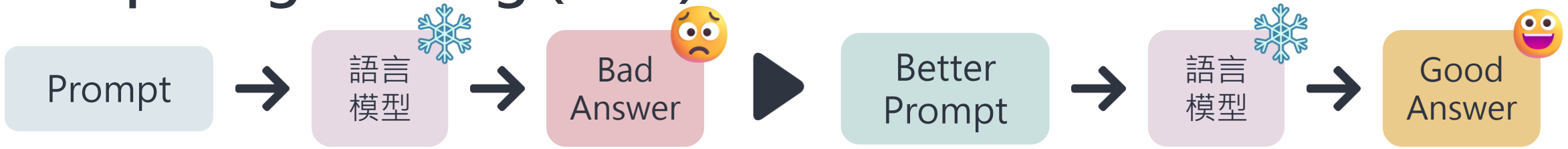
Retrieval-Augmented Generation 檢索增強生成



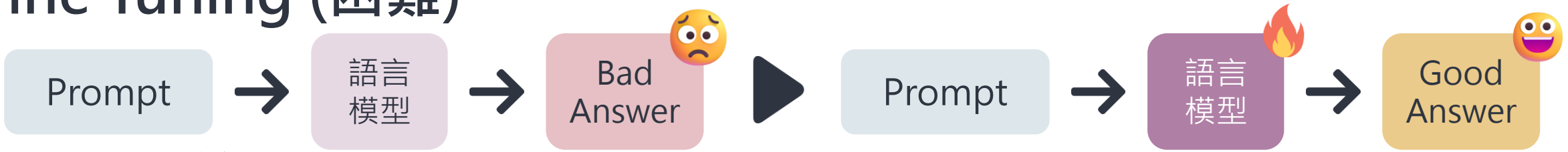
03

提升模型表現方式

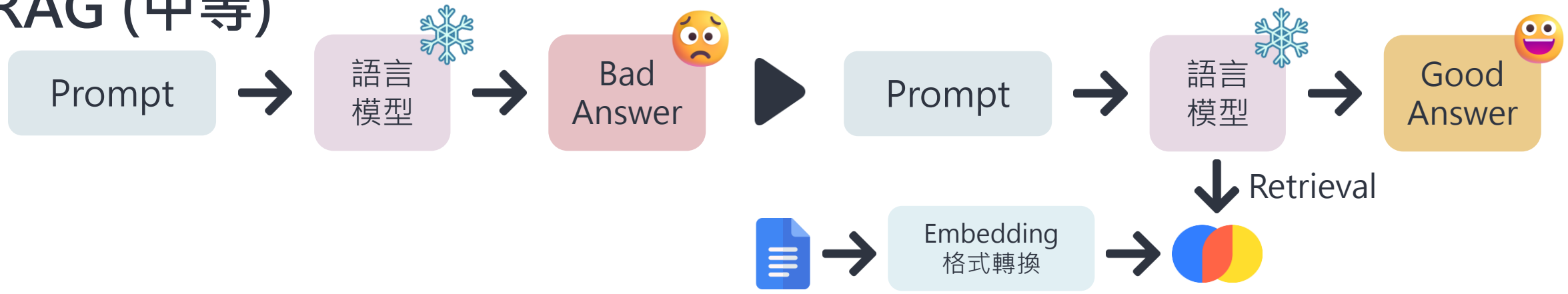
Prompt Engineering (簡單)



Fine Tuning (困難)

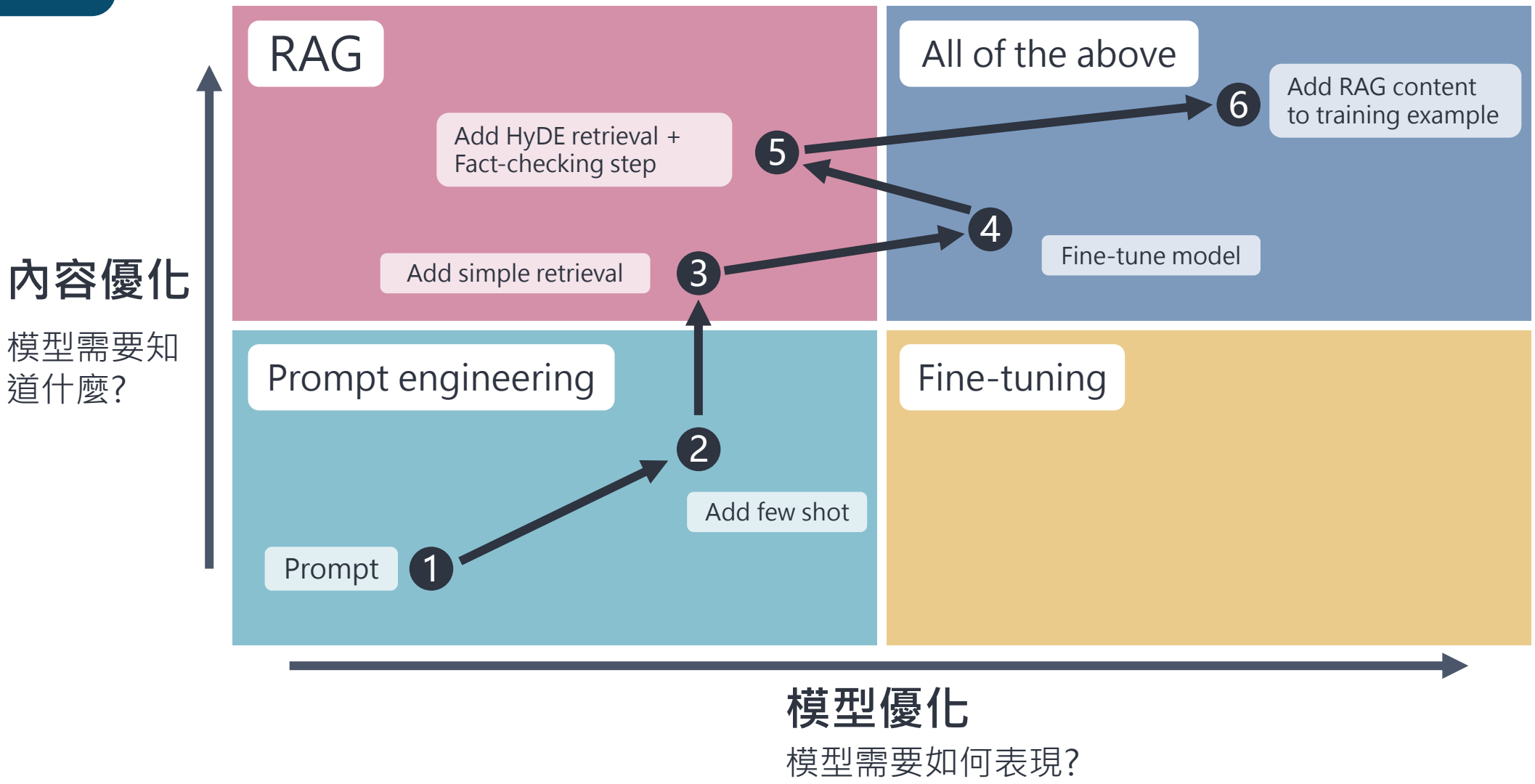


RAG (中等)



03

最佳化策略


RAG
All of the above
Add HyDE retrieval +
Fact-checking step
5
6
Add RAG content
to training example
4
Fine-tune model
Add simple retrieval
3
Prompt engineering
Fine-tuning
2
Add few shot
Prompt
1
內容優化
模型需要知
道什麼?
模型優化
模型需要如何表現?

04

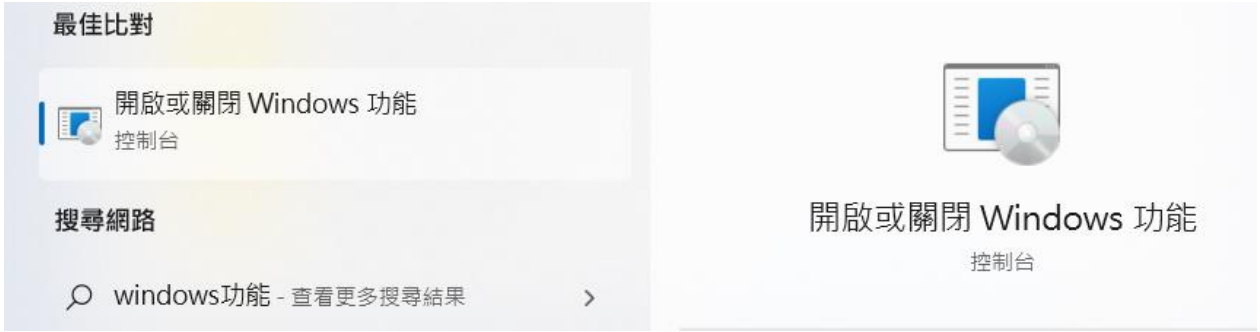
Open webUI

Open webUI 是一種基於 Web 的使用者介面，用於管理和操作各種本地和雲端的人工智慧模型。它提供了一個直覺的圖形化介面(類似Chat GPT介面)，使用戶可以輕鬆地載入、配置、運行和監控各種 AI 模型，而無需編寫程式碼或使用命令列介面。

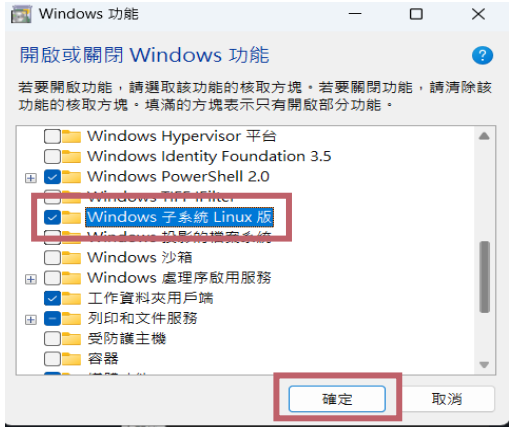
04

Open webUI 安裝

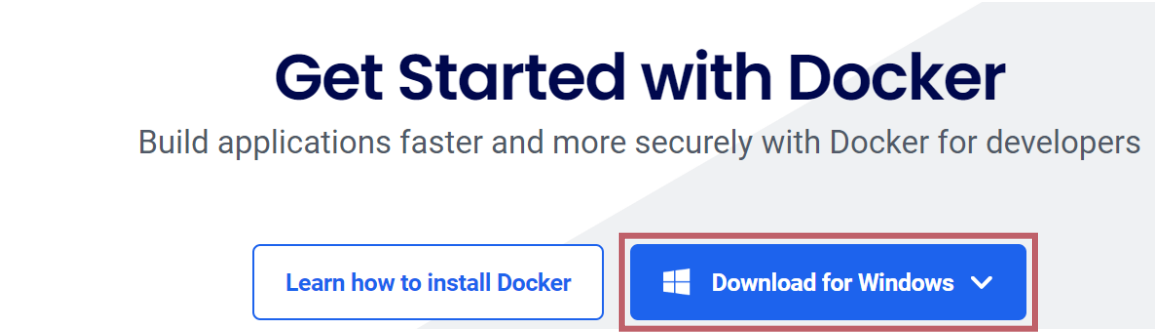
1. 點選搜尋列(Win+S)找到開啟或關閉Windows功能



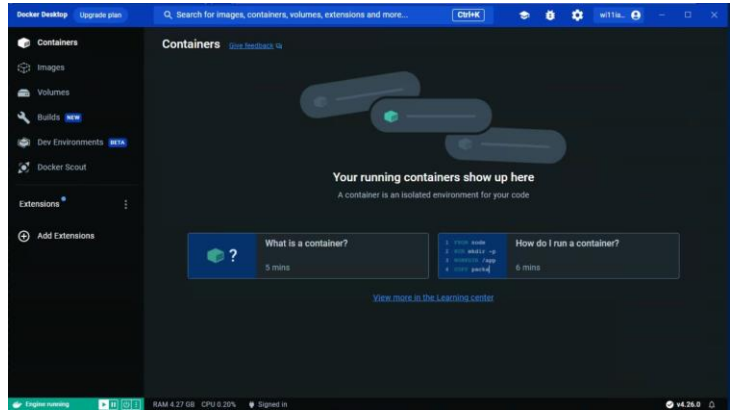
2. 找到Windows子系統Linux版點擊打勾



3. 進入Docker 網站並下載並安裝 https://www.docker.com/get-started/



4. 出現如右圖程式化面即成功安裝！！



04

Open webUI 安裝

5. 在cmd中輸入 **docker run -d -p 3000:8080 --gpus all --add-host=host.docker.internal:host-gateway -v <自己的備份位置>:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:cuda** 執行webui

指令	指令說明
docker run	運行指令
-p <外部端口>:<內部端口>	將容器外的端口接入容器內 為了連線所需
-d	背景運行
--gpus all	GPU 使用權限 這裡設定為所有GPU
--add-host=...	增加宿主(常規寫法)
-v <本機資料位置>:<容器資料位置>	將內部重要資料 備份到外部資料夾 c:/data:/app/backend/data
--name <容器名稱>	設定容器名稱
--restart always	開機自動重啟
ghcr.io/open-webui/open-webui:cuda	映像檔名稱

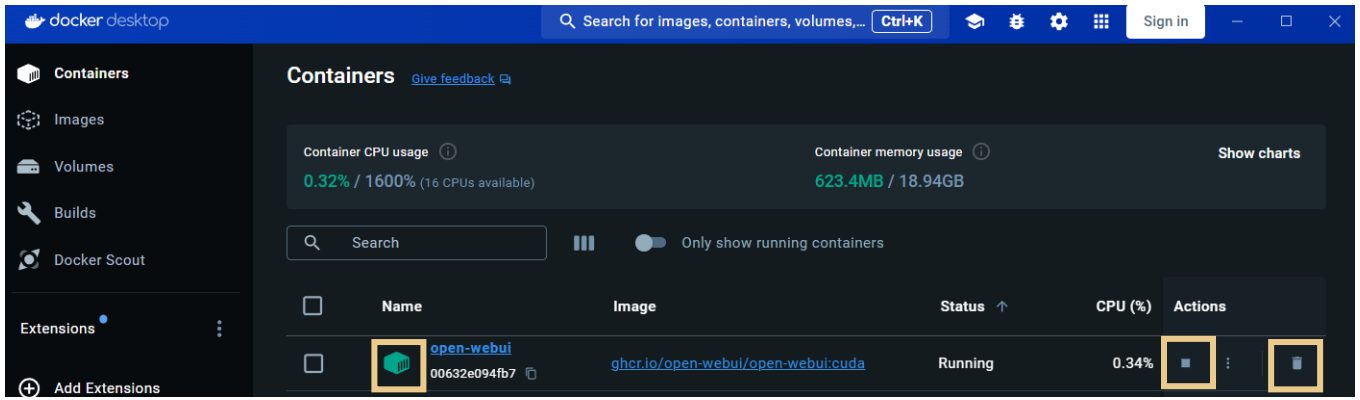
04

Open webUI 安裝

6. 開啟成功下方會出現一些數值代表開啟成功

```
❯  docker run -d -p 3000:8080 --gpus all --add-host=host.docker.intern
al:host-gateway -v c:/workspce/open-webui:/app/backend/data --name ope
n-webui --restart always ghcr.io/open-webui/open-webui:cuda
00632e094fb7db9b300026990a13b4aa38cd55733b778272292cb54887f9fe6a
```

7. 打開Docker Desktop 在 Containers 中有 open-webui 容器



綠色 正在運行
灰色 停止運行

開始/結束運行 刪除容器

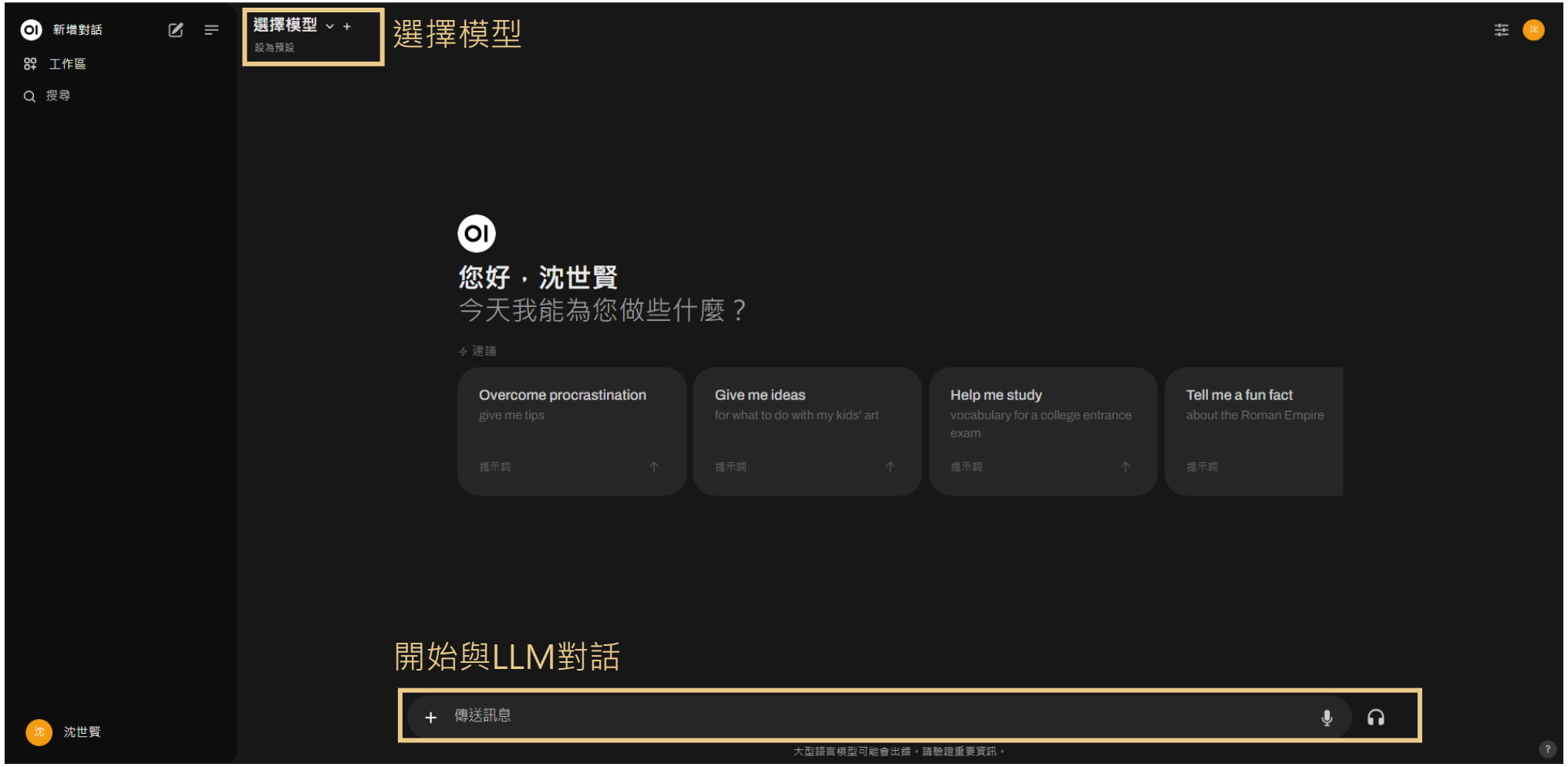
8. 開啟瀏覽器輸入 http://localhost:3000/

9. 註冊帳號(第一個默認為管理員)，然後登入

04

Open webUI 使用

10. 開始使用屬於自己的LLM！！



05

RAG 模型實作

Jupyter 啟動！！