40 分鐘簡單聊聊 NLP

講者：Andy Chiang

自我介紹

- **Andy Chiang（江尚軒）**
- 中興大學資工系 大三升大四
- 中興大學 NLP 實驗室 研究助理
- 工研院 資料服務與智慧決策部 實習生
- 主要研究領域：網頁前後端、機器學習和自然語言處理


再當一年聽眾
跑來當講者

開始之前...

投影片連結

議程資訊

目錄

1 What? NLP 簡介

2 Why? NLP 實際應用

3 When? NLP 發展史

4 How?
Google Colab +
Hugging Face 實作

5 Reference

6 Summary

1

What?
NLP 簡介

40 分鐘簡單聊聊 NLP

What? NLP 簡介

自然語言處理

(Natural Language Processing, NLP)

= 電腦科學 + 語言學

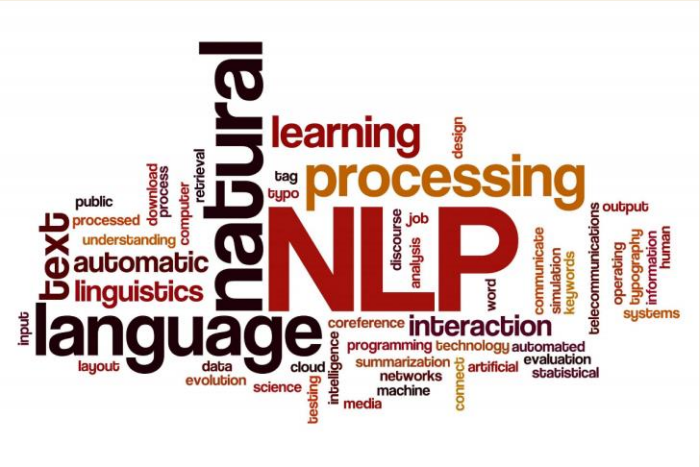
What? NLP 簡介

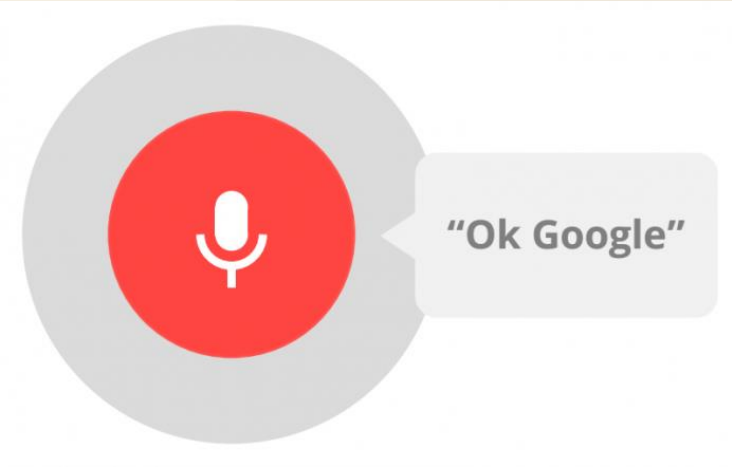
探討如何讓電腦理解、運用自然語言。

What? NLP 簡介

Q：什麼是**自然語言**?

A：人類為了溝通所創造的語言，形式可以是**文字**、**語音**、**符號**...





What? NLP 簡介

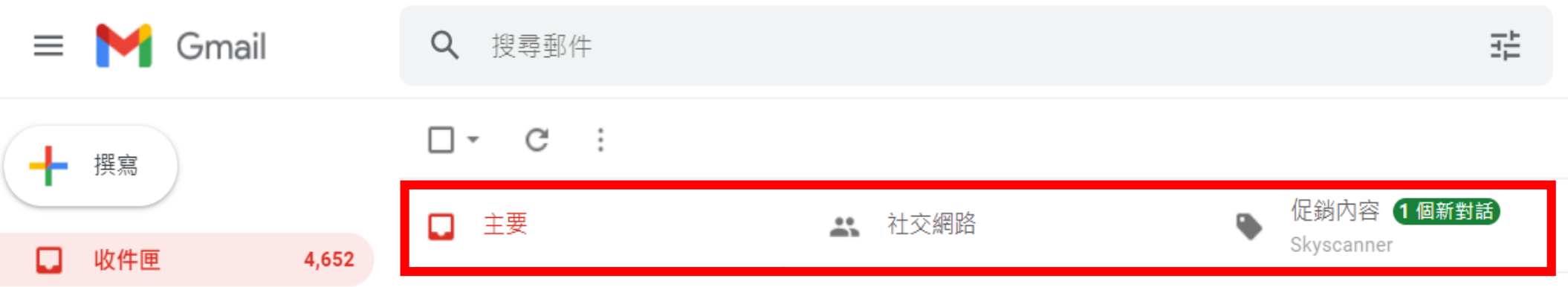
2

Why?
NLP 實際應用

Why? NLP 實際應用

Email 篩選器

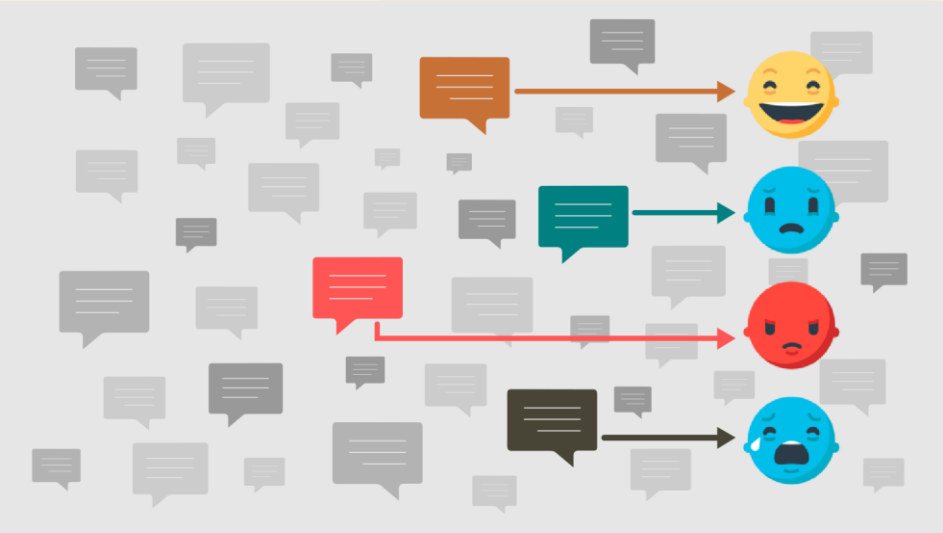
- 過濾**垃圾郵件**
- Gmail 信件分類



Why? NLP 實際應用

情感分析

- 蒐集客戶對該產品的相關留言或貼文
- 分析出**正面**或**負面**的比例



Why? NLP 實際應用

智能助理

- 處理生活瑣事
- 聊天、講笑話~

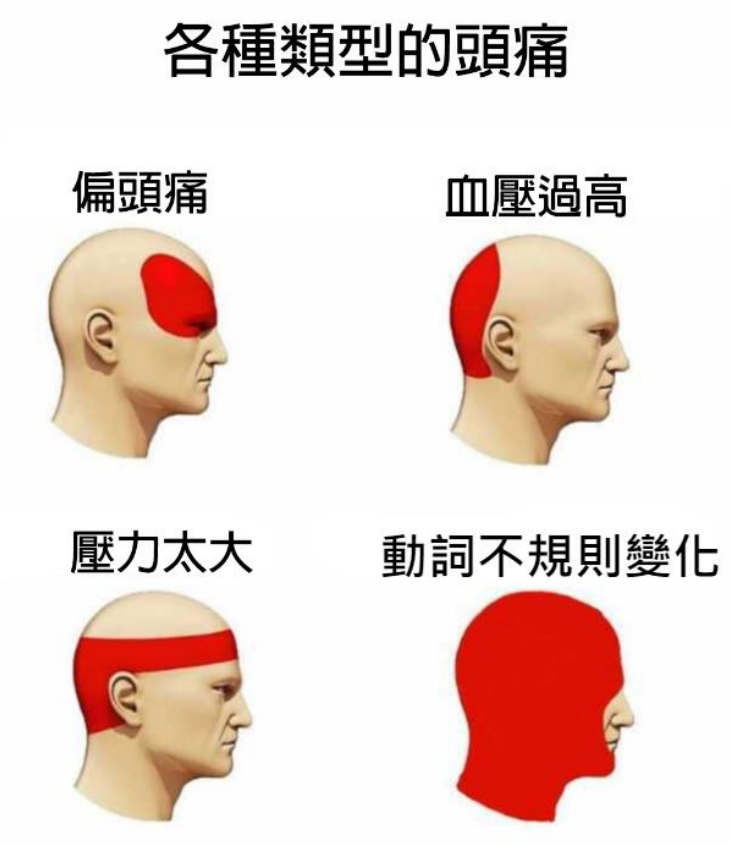
3

When?
NLP 發展史

When? NLP 發展史

人工規則

在 **1950** 年代，當時只能透過**語言學分析**語言的規則後，再寫成**電腦程式**。

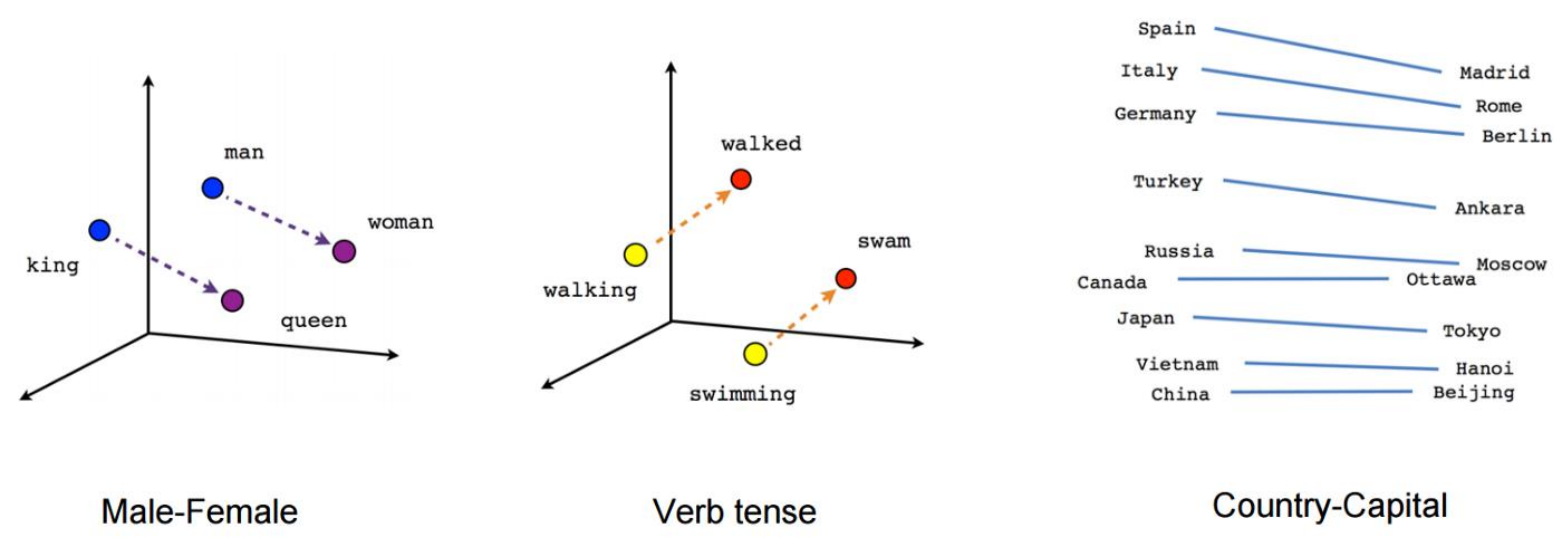


When? NLP 發展史

直到 **1980** 年代，NLP 開始代入**機器學習**的概念。

When? NLP 發展史

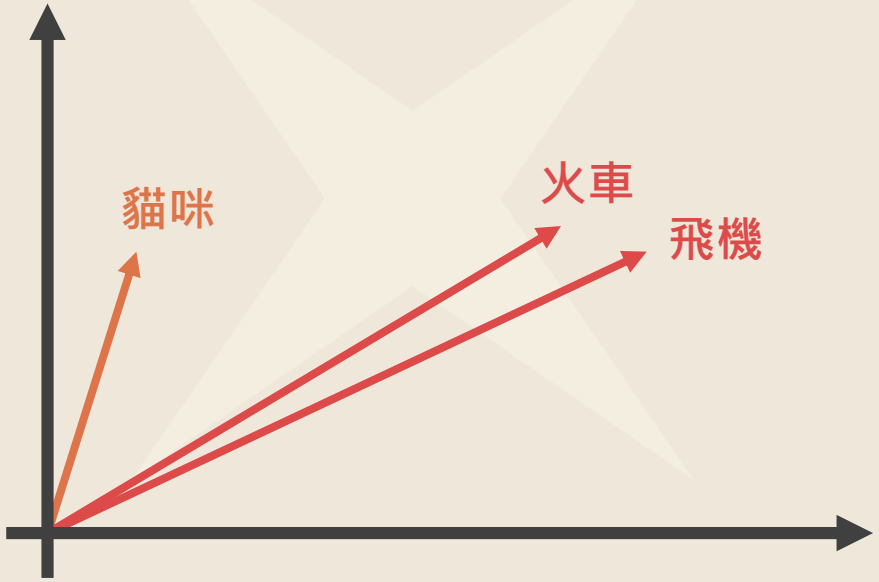
Word2Vec 的輸入是**單字**，輸出就是代表此**單字的向量**。



When? NLP 發展史

我明天要搭火車去台北

我明天要搭飛機去台北



When? NLP 發展史

Word2Vec 的缺點：

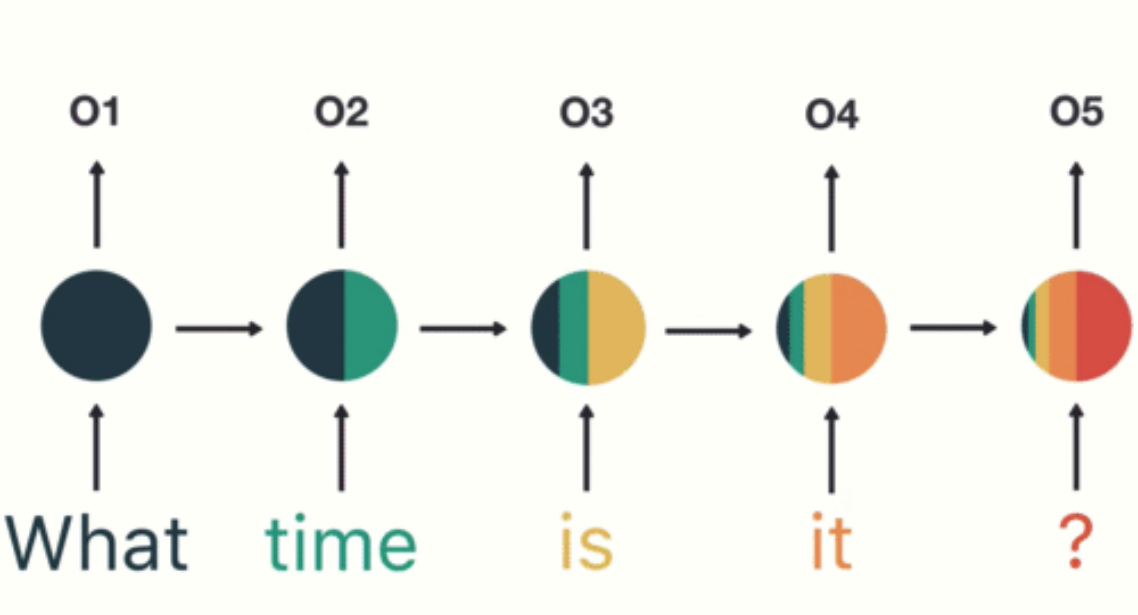
1. 無法處理**一字多義**
2. 不考慮詞的**先後順序**

我**借**小明100元

小明**借**我100元

When? NLP 發展史

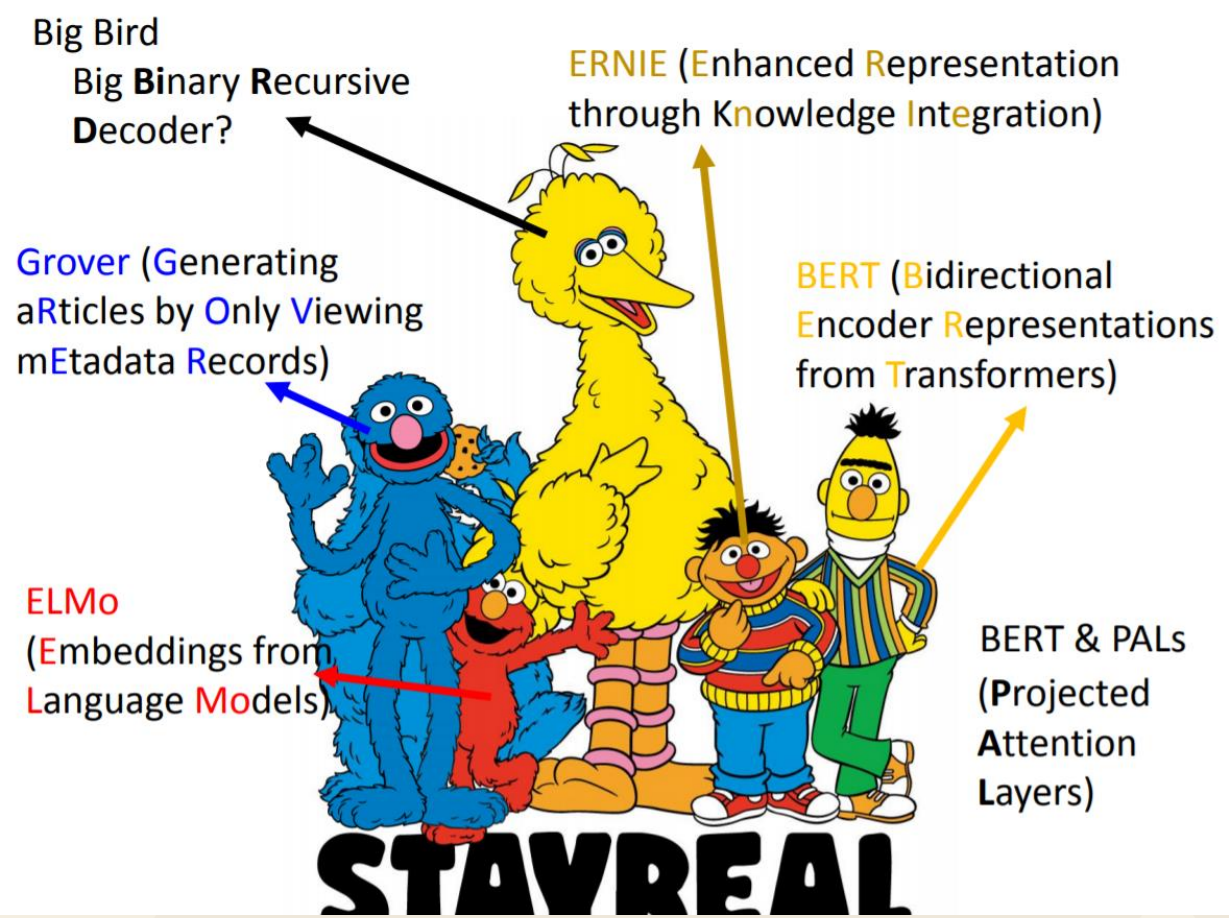
RNN 就像我們閱讀，是**從左開始一個一個字讀過來**。



When? NLP 發展史

自從 2018 年 Google 提出 BERT 語言預訓練模型後，對 NLP 帶來革命性的突破。

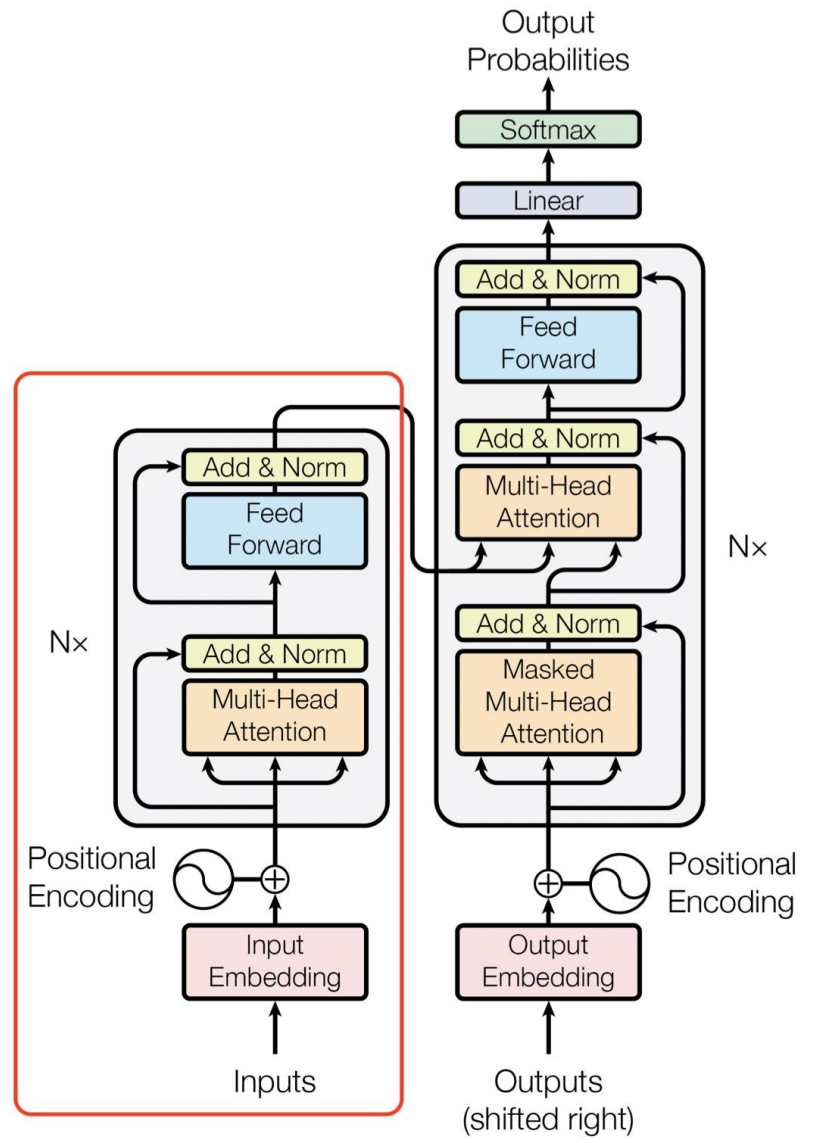
When? NLP 發展史


Big Bird
Big Binary Recursive Decoder?
ERNIE (Enhanced Representation through Knowledge Integration)
Grover (Generating aRticles by Only Viewing mEtadata Records)
BERT (Bidirectional Encoder Representations from Transformers)
ELMo (Embeddings from Language Models)
BERT & PALs (Projected Attention Layers)
STAYREAL

When? NLP 發展史

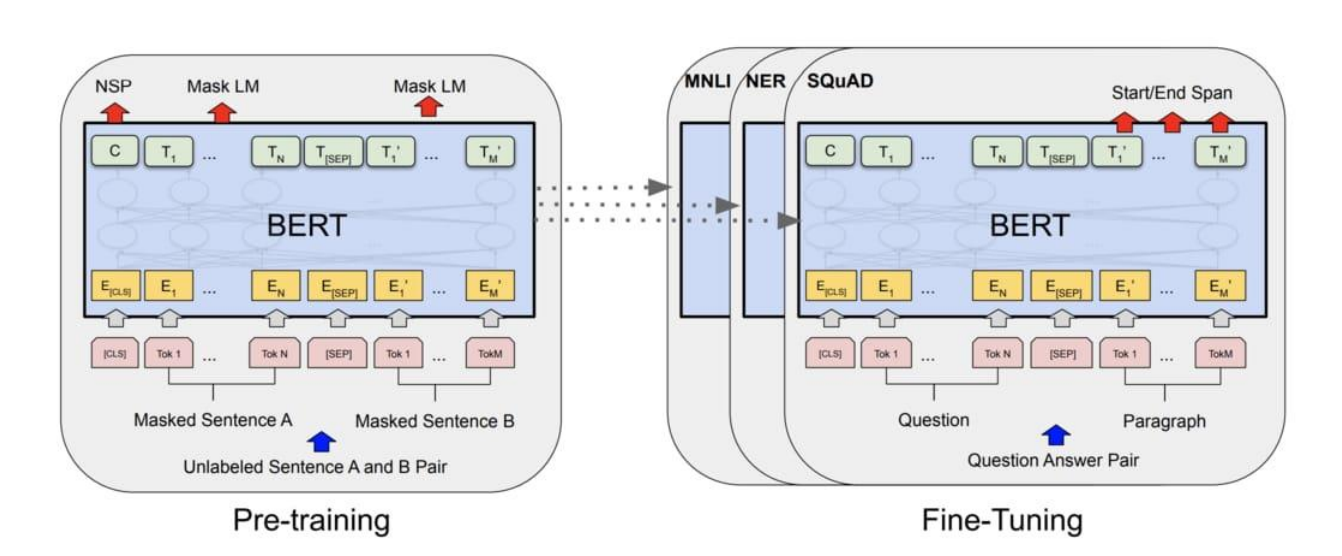
Transformer

- Self-attention
- **Encoder** & Decoder



When? NLP 發展史

- 大量的**語料庫**
- 透過**非監督**的方式來 **pre-train**
- 對特定的下游任務作 **fine-tune**



When? NLP 發展史

這概念其實就像我們學中文一樣。

pre-train >> 學會基本語感

fine-tune >> 學習特定的任務

When? NLP 發展史

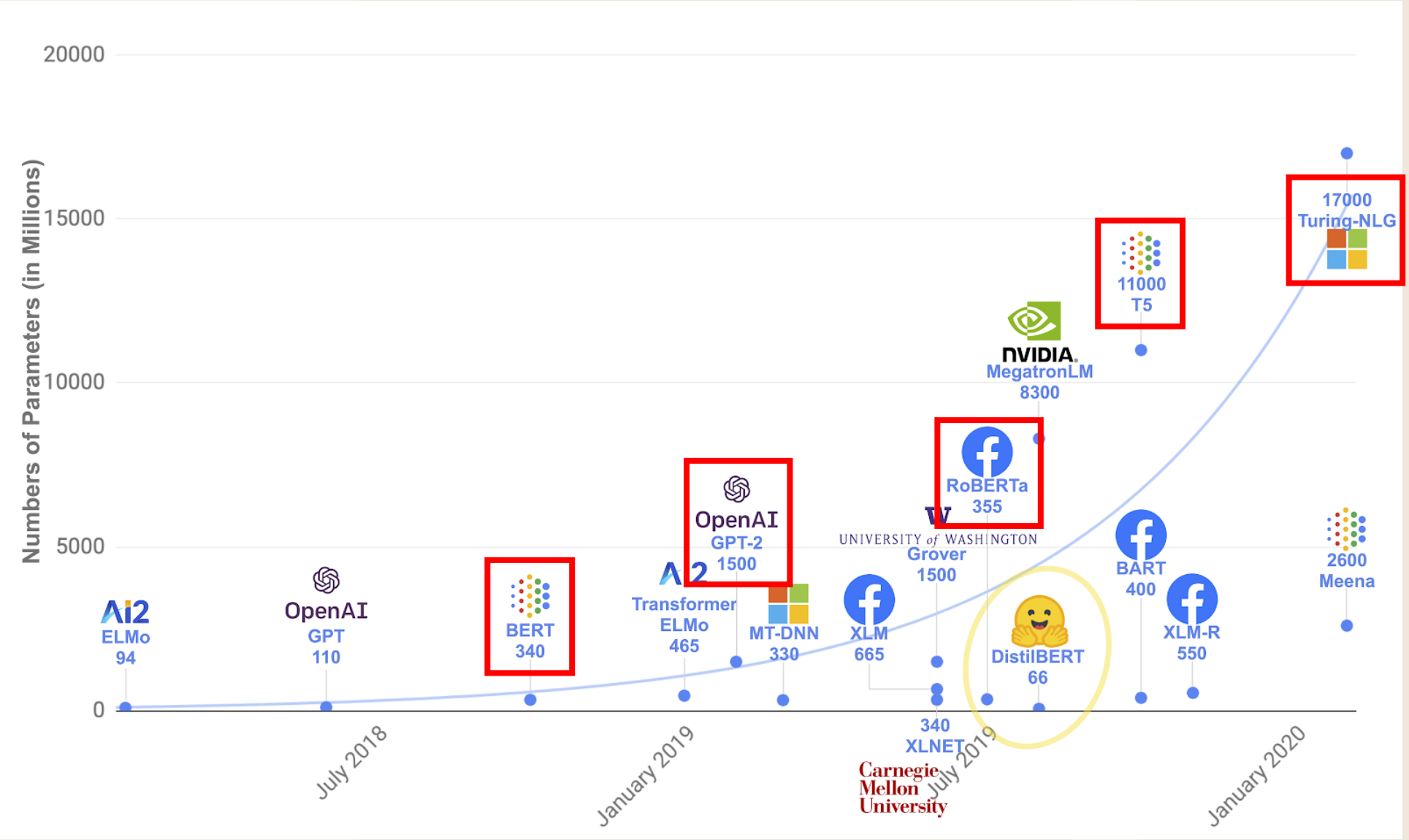
但除非是大企業，不然要自己 pre-train 語言預訓練模型根本是天方夜譚。

幸好 BERT 作者有開源 pre-train 好的模型，**讓我們可以直接站在巨人的肩膀上，讓下游任務變得既輕鬆又有效**。

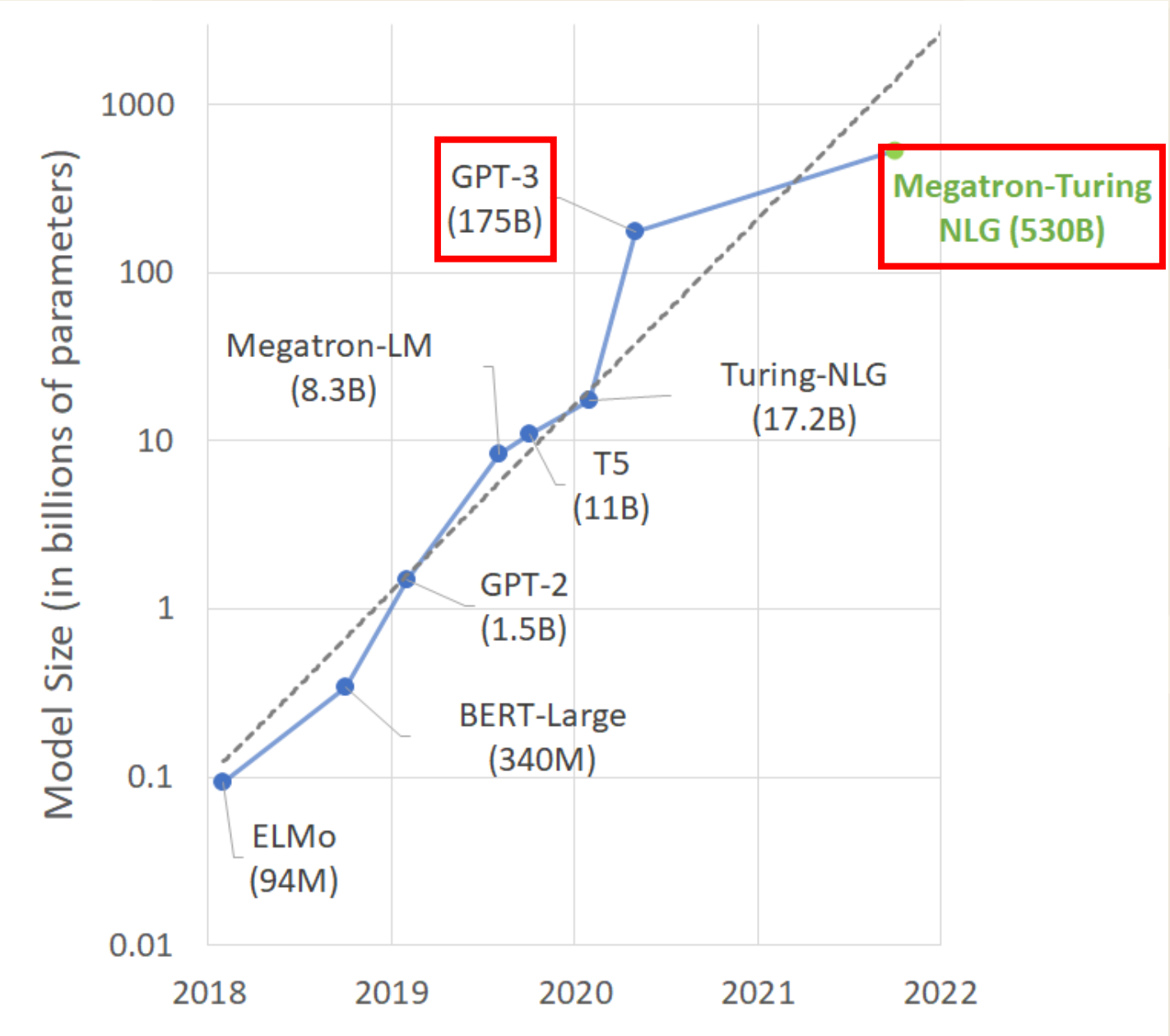

NLP研究者
BERT
傳統模型

When? NLP 發展史

Google 提出了 BERT 後，其他大企業或組織也競相推出了自己的語言預訓練模型。



40 分鐘簡單聊聊 NLP


Model Size (in billions of parameters)
1000
100
10
1
0.1
0.01
2018
2019
2020
2021
2022
GPT-3
(175B)
Megatron-Turing
NLG (530B)
Megatron-LM
(8.3B)
Turing-NLG
(17.2B)
T5
(11B)
GPT-2
(1.5B)
BERT-Large
(340M)
ELMo
(94M)

4

How?
Google Colab + Hugging Face 實作

How? Google Colab + Hugging Face 實作

首先介紹 **Google Colab**，底下列出一些優缺點：

How? Google Colab + Hugging Face 實作

優點：

- **不需要架設環境**
- 內建許多**機器學習**的套件
- **免費使用 GPU、TPU**
- 容易**分享**、**共用**
- **視覺化**呈現執行結果

How? Google Colab + Hugging Face 實作

缺點：

- 連續運行時間最長為 **12 小時**
- **重啟資料會被清除**
- **GPU、TPU 有用量限制**

How? Google Colab + Hugging Face 實作

Hugging Face

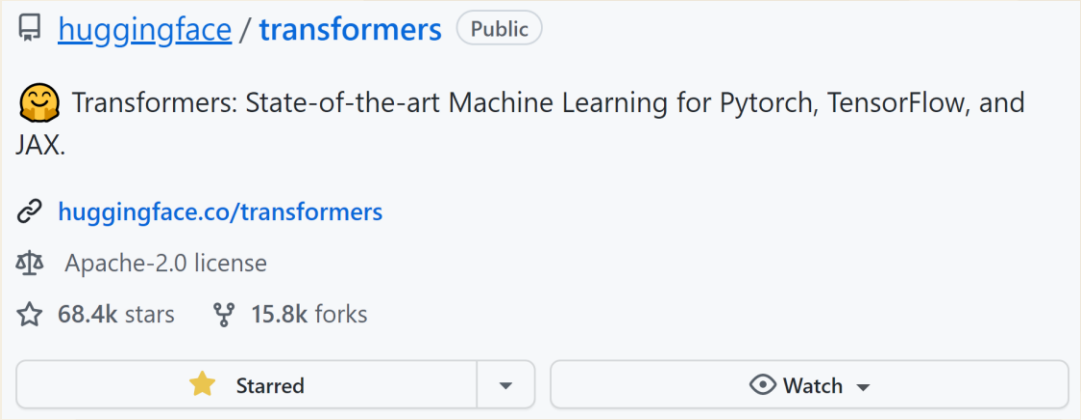
- 讓最先進的 NLP 模型易於使用
- 釋出很多的**語言預訓練模型**
- 網站很好看XD

Hugging Face

How? Google Colab + Hugging Face 實作

Transformers

- **下載**、**訓練**、**上傳**語言預訓練模型
- 文本分類、文本生成、問答等任務
- **GitHub** 上有 **6.8 萬個 star**



How? Google Colab + Hugging Face 實作

Colab連結

溫馨小提醒：

1. 執行前請記得另存一個新的副本才能執行哦！
2. 變更執行階段類型，改成 GPU，不然訓練會慢到崩潰哦！
3. 點「連線」後就可以開始執行了。

How? Google Colab + Hugging Face 實作

Install packages

- 下載 transformers 和 datasets 套件

```
!pip install transformers datasets
```

How? Google Colab + Hugging Face 實作

Pipeline

- 執行 NLP 任務最簡單的工具

How? Google Colab + Hugging Face 實作

情感分析 (sentiment-analysis)

```
from transformers import pipeline

classifier = pipeline("sentiment-analysis")
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english
```

How? Google Colab + Hugging Face 實作

正面

```
classifier("This movie is interesting!")
```

```
[{'label': 'POSITIVE', 'score': 0.99983811378479}]
```

負面

```
classifier("This movie is boring!")
```

```
[{'label': 'NEGATIVE', 'score': 0.9998012185096741}]
```

How? Google Colab + Hugging Face 實作

文本生成 (text-generation)

```
generator = pipeline("text-generation")
```

```
No model was supplied, defaulted to gpt2
```

How? Google Colab + Hugging Face 實作

```
generator("Once upon a time there were three little pigs.")
```

```
[{'generated_text': "Once upon a time there were three little pigs. She was pregnant with two, one her father and one his mother. The baby boy turned five months old on the day of the baby's birth. He and his mother got divorced. He had to"}]
```

How? Google Colab + Hugging Face 實作

問答 (question-answering)

```
qa = pipeline("question-answering")
```

```
No model was supplied, defaulted to distilbert-base-cased-distilled-squad
```

How? Google Colab + Hugging Face 實作

```
context = """
Harry Potter is a series of seven fantasy novels
The novels chronicle the lives of a young wizard,
all of whom are students at Hogwarts School of Wi
Harry's struggle against Lord Voldemort, a dark wiz
governing body known as the Ministry of Magic and
"""
question = "Who is the author of Harry Potter?"

qa(question, context)
```

```
'answer': 'J. K. Rowling'
```

How? Google Colab + Hugging Face 實作

使用自訂模型

- Hugging Face 網站的 models 頁面

Hugging Face | Search models, dataset: | Models | Datasets | Spaces | Docs | Solutions | Pricing

Tasks

- Image Classification
- Translation
- Image Segmentation
- Fill-Mask
- Automatic Speech Recognition
- Token Classification
- Sentence Similarity
- Audio Classification
- Question Answering
- Summarization
- Zero-Shot Classification

+ 18 Tasks

Libraries

- PyTorch
- TensorFlow
- JAX

+ 28

Datasets

- wikipedia
- common_voice
- squad
- glue

Models 66,100

Filter by name | Sort: Most Downloads

bert-base-uncased
Fill-Mask • Updated Jun 6 • ↓ 27.2M • ♥ 228

microsoft/deberta-base
Updated Jan 13 • ↓ 21.8M • ♡ 24

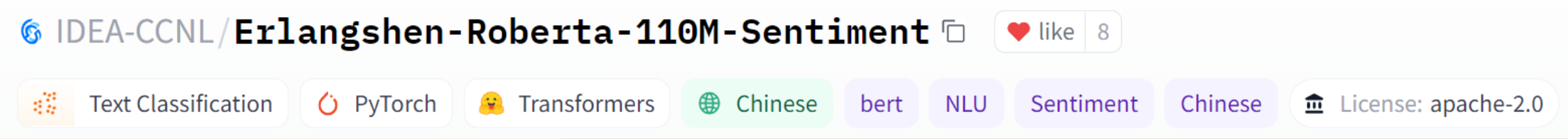
Jean-Baptiste/camembert-ner
Token Classification • Updated Apr 4 • ↓ 19.5M • ♡ 14

gpt2
Text Generation • Updated May 20, 2021 • ↓ 13.1M • ♡ 188

bert-base-cased
Fill-Mask • Updated Sep 6, 2021 • ↓ 10.6M • ♡ 37

How? Google Colab + Hugging Face 實作

- **Tasks**：Text Classification
- **Languages**：Chinese
- **Model**：IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment

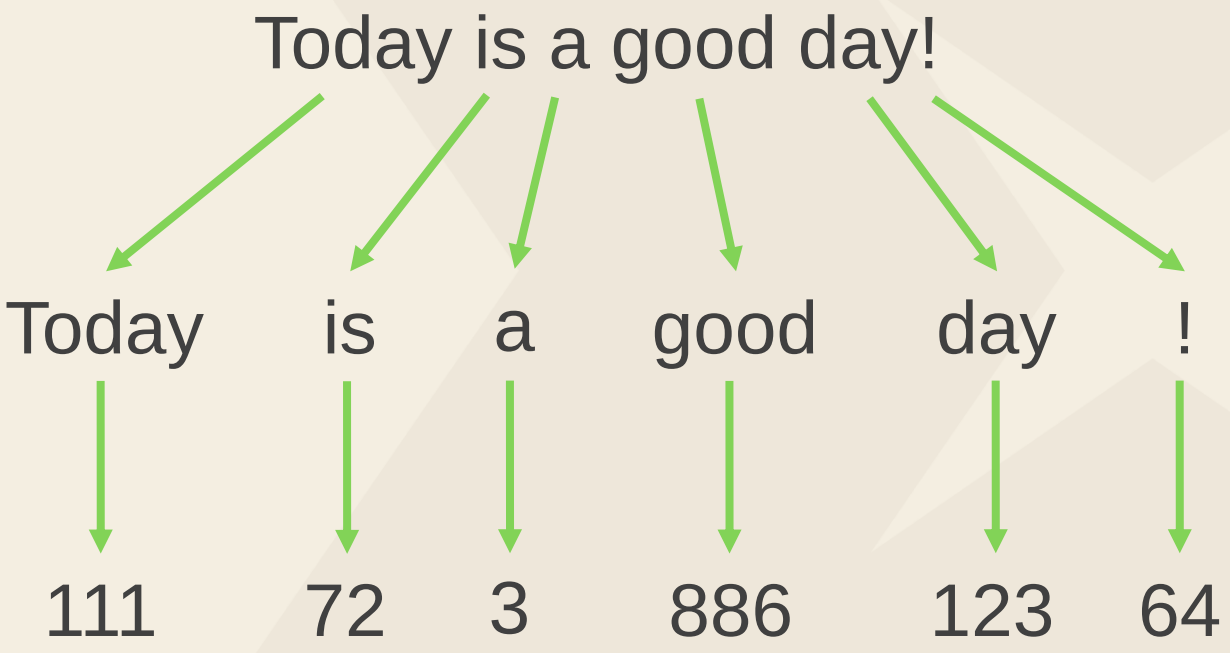


How? Google Colab + Hugging Face 實作

- **AutoModelForSequenceClassification**：指定任務的模型
- **AutoTokenizer**：文字轉數字 (token)

```
from transformers import AutoModelForSequenceClassification, AutoTokenizer

model = AutoModelForSequenceClassification.from_pretrained("IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment")
tokenizer = AutoTokenizer.from_pretrained("IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment")
```

How? Google Colab + Hugging Face 實作



How? Google Colab + Hugging Face 實作

```
from transformers import pipeline

classifier = pipeline("sentiment-analysis", model=model, tokenizer=tokenizer)
```

正面

```
classifier("這部電影真有趣!")
```

```
[{'label': 'Positive', 'score': 0.7309864163398743}]
```

負面

```
classifier("這部電影真無聊!")
```

```
[{'label': 'Negative', 'score': 0.9995669722557068}]
```

How? Google Colab + Hugging Face 實作

How? Google Colab + Hugging Face 實作

Datasets

- Hugging Face 網站的 datasets 頁面

Hugging Face | Search models, datasets | Models | Datasets | Spaces | Docs | Solutions | Pricing

Task Categories

text-classification | question-answering | text-generation
token-classification | translation | text2text-generation
+ 140 Task Categories

Tasks

language-modeling | multi-class-classification
extractive-qa | named-entity-recognition
open-domain-qa | natural-language-inference | + 372

Languages

English | German | French | Spanish
Russian | Arabic | + 182

Datasets 8,721 | Filter by name | Sort: Most Downloads

red_caps
Preview • Updated Jul 1 • ↓ 6.41M • ♡ 12

allenai/nllb
Preview • Updated 5 days ago • ↓ 2.77M • ♡ 1

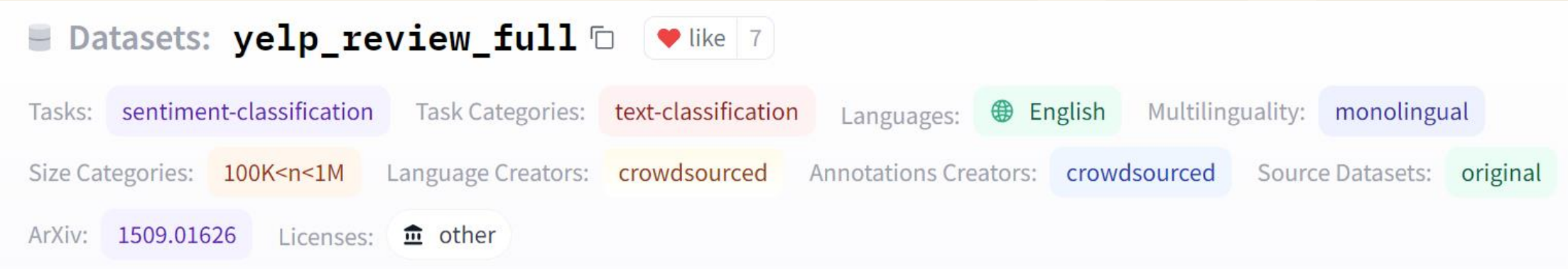
super_glue
Preview • Updated about 21 hours ago • ↓ 1.49M • ♡ 23

glue
Preview • Updated 11 days ago • ↓ 1.26M • ♡ 51

GEM/wiki_lingua

How? Google Colab + Hugging Face 實作

- **Task Categories**：text-classification
- **Tasks**：sentiment-classification
- **Languages**：English
- **Dataset**：yelp_review_full



How? Google Colab + Hugging Face 實作

下載資料集

```
1 from datasets import load_dataset
2
3 dataset = load_dataset("yelp_review_full")
```

How? Google Colab + Hugging Face 實作

查看其中一筆資料

```
dataset["train"][0]
```

```
{'label': 4,
 'text': "dr. goldberg offers everything i look for in a general practitioner.  he's nice and easy to talk to without being patronizing; he's always on time in seeing his patients; he's affiliated with a top-notch hospital (nyu) which my parents have explained to me is very important in case something happens and you need surgery; and you can get referrals to see specialists without having to see him first.  really, what more do you need?  i'm sitting here trying to think of any complaints i have about him, but i'm really drawing a blank."}
```

How? Google Colab + Hugging Face 實作

Tokenizer

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
```

How? Google Colab + Hugging Face 實作

資料預處理

```
def tokenize_function(examples):
    return tokenizer(examples["text"], padding=True, truncation=True)
```

```
tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

How? Google Colab + Hugging Face 實作

Fine-tune 模型

- bert-base-cased

```
from transformers import AutoModelForSequenceClassification

model = AutoModelForSequenceClassification.from_pretrained("bert-base-cased", num_labels=5)
```

How? Google Colab + Hugging Face 實作

衡量指標

```
import numpy as np
from datasets import load_metric

metric = load_metric("accuracy")
```

```
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels)
```

How? Google Colab + Hugging Face 實作

超參數

```
from transformers import TrainingArguments

training_args = TrainingArguments(output_dir="test_trainer", evaluation_strategy="epoch")
```

How? Google Colab + Hugging Face 實作

Trainer

```
from transformers import Trainer

trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=tokenized_datasets["train"],
        eval_dataset=tokenized_datasets["test"],
        compute_metrics=compute_metrics,
)
```

How? Google Colab + Hugging Face 實作

Fine-tune

```
trainer.train()
```

[1500/1500 22:17, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy
1	1.307200	0.988094	0.577000
2	0.936900	0.970908	0.596000
3	0.621400	1.052324	0.606000

How? Google Colab + Hugging Face 實作

儲存模型

```
trainer.save_model("./test_model")
```

How? Google Colab + Hugging Face 實作

載入模型

```
from transformers import AutoModelForSequenceClassification

pt_model = AutoModelForSequenceClassification.from_pretrained("./test_model")
```

```
from transformers import TFAutoModelForSequenceClassification

tf_model = TFAutoModelForSequenceClassification.from_pretrained("./test_model", from_pt=True)
```

How? Google Colab + Hugging Face 實作

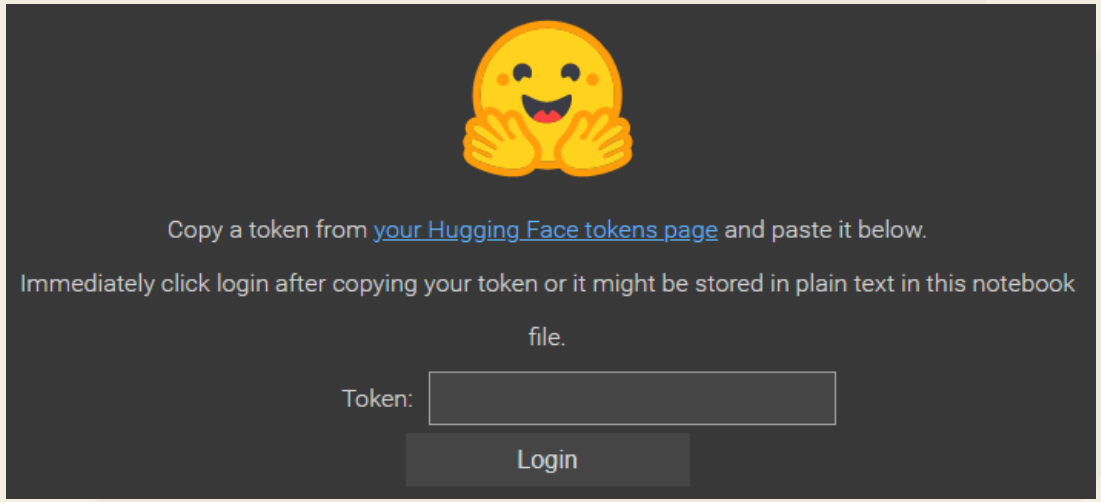
分享模型

```
!pip install huggingface_hub
```

How? Google Colab + Hugging Face 實作

分享模型

```
from huggingface_hub import notebook_login

notebook_login()
```



How? Google Colab + Hugging Face 實作

分享模型

```
pt_model.push_to_hub("my-test-model")
```

```
tf_model.push_to_hub("my-test-model")
```

```
tokenizer.push_to_hub("my-test-model")
```

How? Google Colab + Hugging Face 實作

下載模型

```
from transformers import AutoModelForSequenceClassification, AutoTokenizer

my_model = AutoModelForSequenceClassification.from_pretrained("AndyChiang/my-test-model")
my_tokenizer = AutoTokenizer.from_pretrained("AndyChiang/my-test-model")
```

How? Google Colab + Hugging Face 實作

下載模型

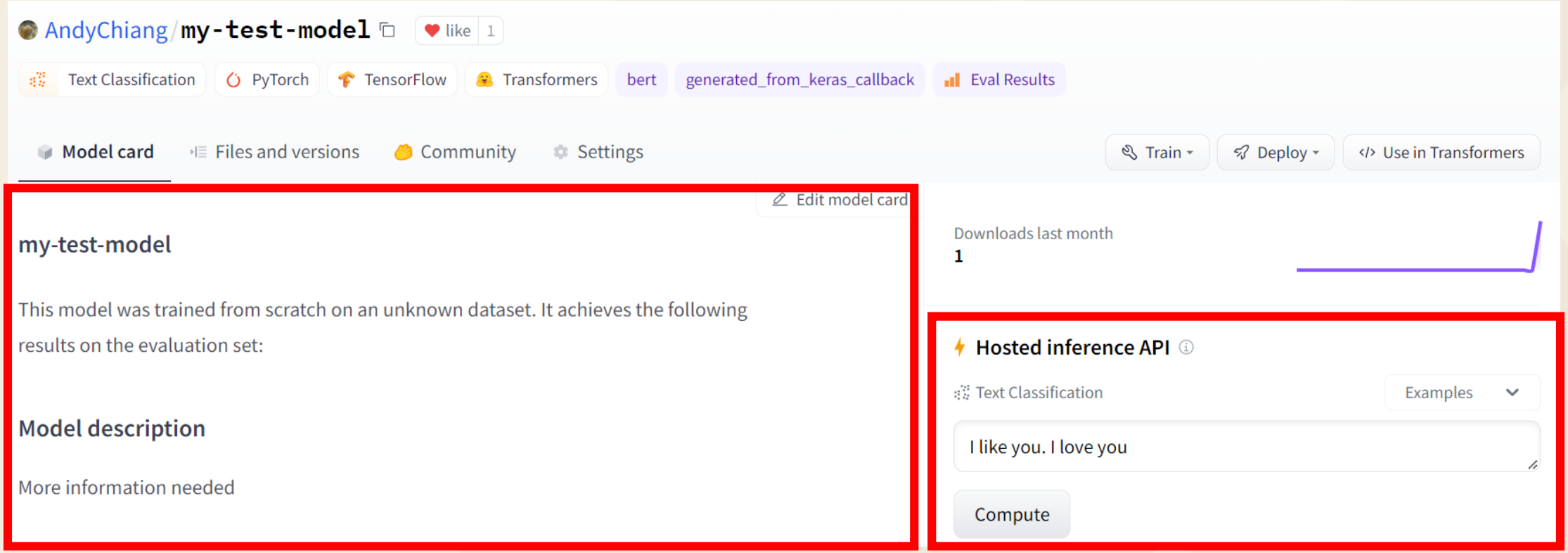
```
from transformers import pipeline

classifier = pipeline("sentiment-analysis", model=my_model, tokenizer=my_tokenizer)
```

```
classifier("This restaurant is great! The food there is delicious, too.")
```

```
[{'label': 'LABEL_4', 'score': 0.9106563329696655}]
```

How? Google Colab + Hugging Face 實作

Hugging Face 網站



How? Google Colab + Hugging Face 實作

Reference

- 斷開中文的鎖鍊！自然語言處理 (NLP)是什麼？
- NLP自然語言處理 – 技術原理與其產業應用
- 進入 NLP 世界的最佳橋樑：寫給所有人的自然語言處理與深度學習入門指南
- 進擊的 BERT：NLP 界的巨人之力與遷移學習
- 台大李宏毅教授 - ELMO, BERT, GPT
- 🤗 Transformers Document

Summary

1. What? NLP 簡介
2. Why? NLP 實際應用
3. When? NLP 發展史
4. How? Google Colab + Hugging Face 實作

CAT IN A MAZE

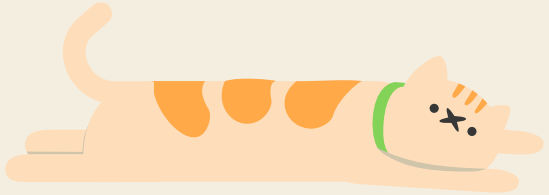
Summary

議程投影片+範例程式碼

個人網頁

Q & A

Slido



Thanks for watching!