www.aiengineer.tw

# AI Study Session

## 111-2 Fine Tune Large Language Models For Your Application

Date: 2023/06/07

Reporter:

Hung-Ming, Lin. Kuan, Yen. Meg, Ho. Guo-Chi, Li. Chien-Yu, Tseng. Chi-Yin, Ho.

# Outlines

1. Introduction of LLM and OpenAI

2. Parameter-Efficient Fine-Tunning (Part I)

Take a Break (10 min.)

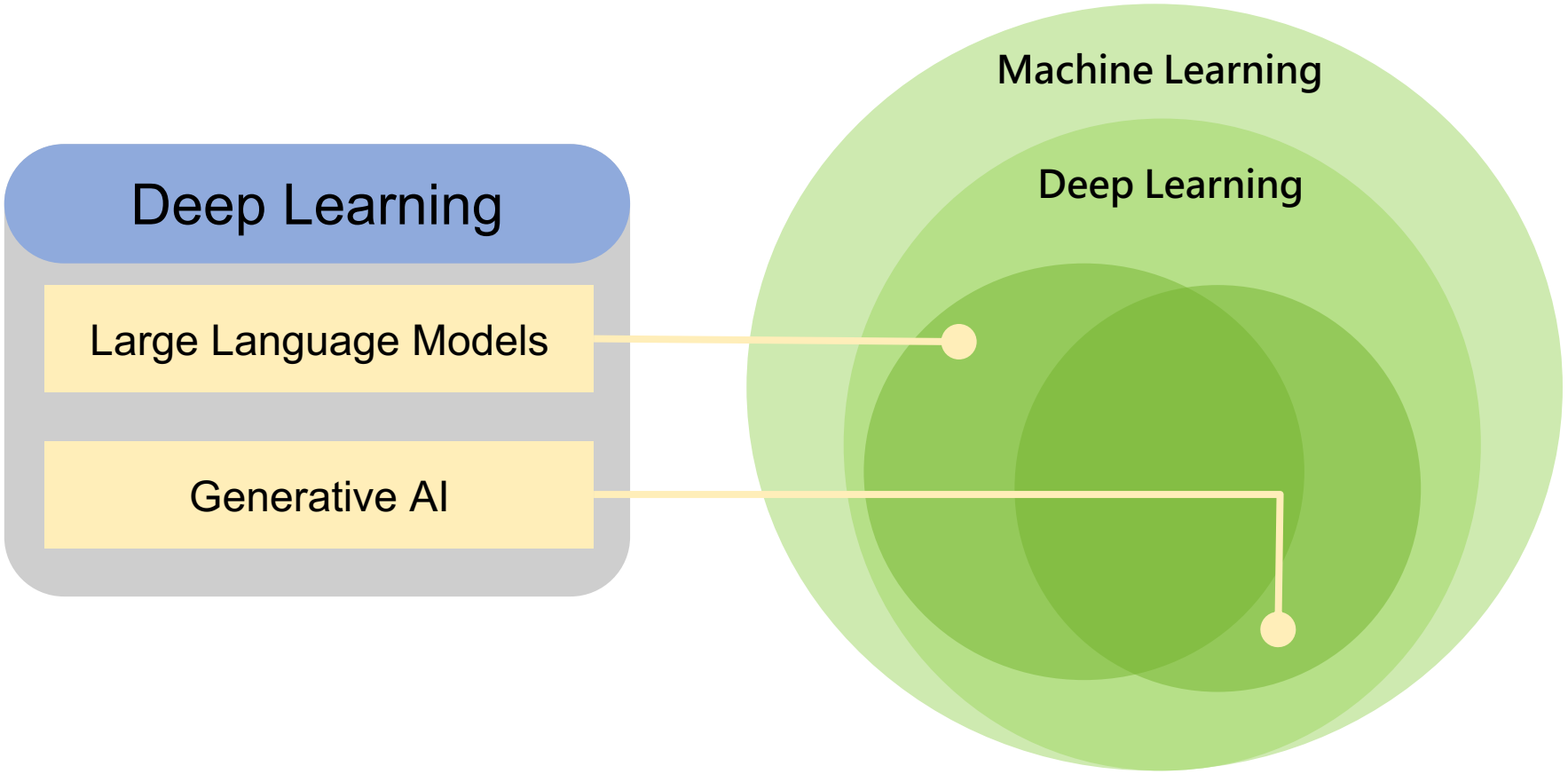
3. Parameter-Efficient Fine-Tunning (Part II)

4. The Application of Word-Embedding: LangChain

Adapted from: Google Cloud

# wedge

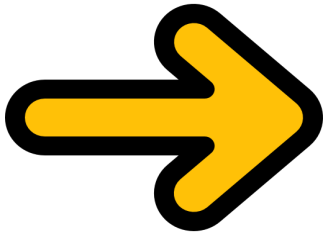
Large Language Models (LLMs) are a subset of Deep Learning




Adapted from: Google Cloud

# wedge

Generative AI



Adapted from: Google Cloud

Image source: flaticon

# What are Large Language Models (LLMs) ?

Adapted from: Google Cloud

Large, general-purpose language models

can be pre-trained and then fine-tuned

for specific purposes

Adapted from: Google Cloud

# wedge

Imaging you are training a dog...

Image source: iStock

# You need more...

+ Special Training

- sniffer dog
- police dog
- search and rescue dog



Image source: 自由時報、泛科學、人間福報、大紀元

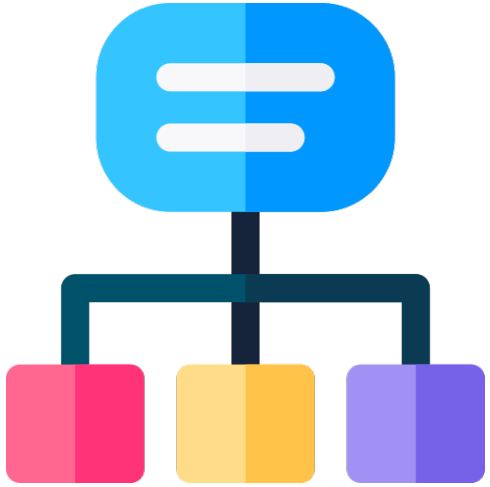
Similar idea applies to

Large Language Models

Adapted from: Google Cloud

# Introduction to LLMs

Large Language Models are trained to solve common language problems, like...



**Text classification**


**Question answering**

**Document summarization**

**Text generation**

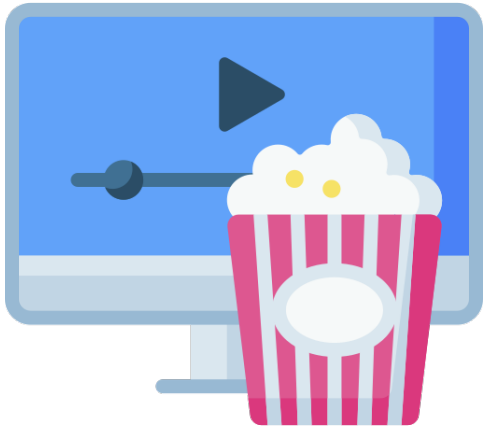
Adapted from: Google Cloud
Image source: flaticon

# Introduction to LLMs

...then be tailored to solve specific problems in different fields, like...


**Retail**

**Finance**



**Entertainment**

**Trained with a relatively small size of field datasets**

Adapted from: Google Cloud
Image source: flaticon

# Introduction to LLMs

## Large Language Models

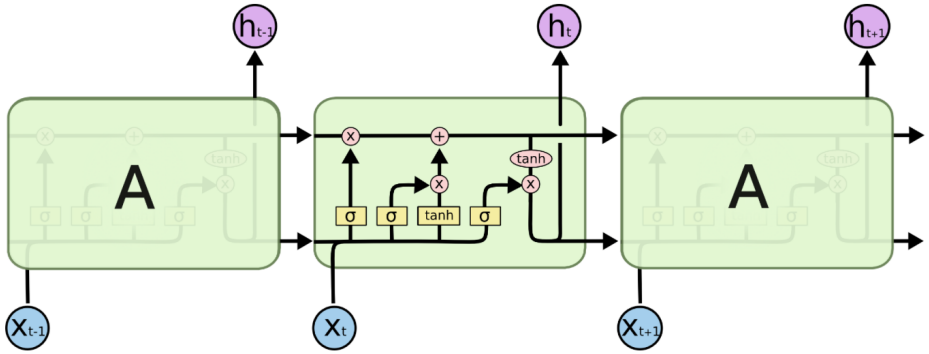
- **Large**
  - Large training dataset [PB]
  - Large number of parameters [B~T]
- **General purpose**
  - Commonality of human languages
  - Resource restriction
- **Pre-trained and fine-tuned**

Adapted from: Google Cloud
Image source: flaticon , U-CAR討論區

# Introduction to LLMs

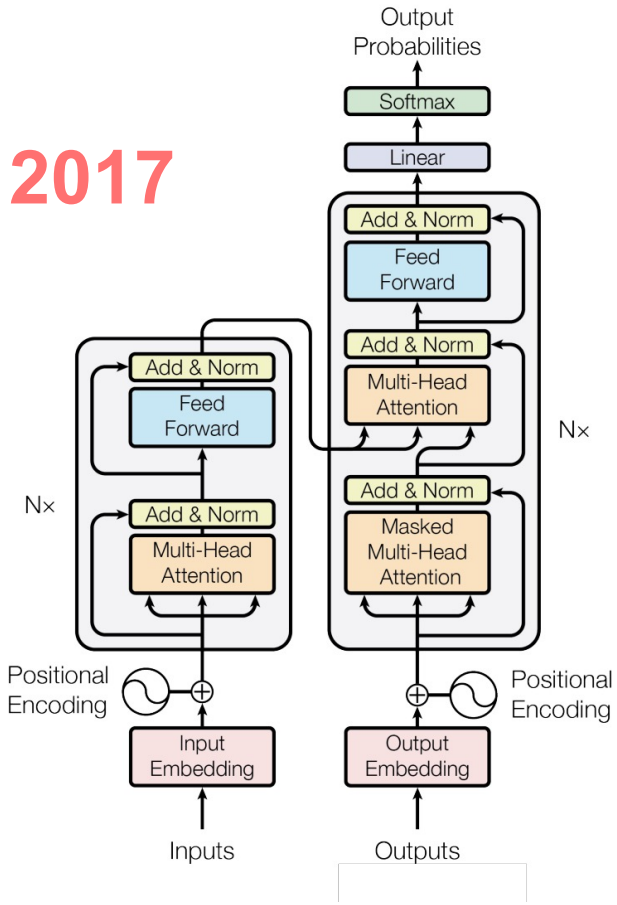
## The turning point

### Before 2017




**RNN-based models (LSTM, GRU …)**

### After 2017




**Transformer**

Image source: Medium , 〈Attention is all you need〉

# Introduction to LLMs

## NTU’s Graduation Speech


NVIDIA CEO Tells NTU Grads：
**Run, Not Walk** — But Be Prepared to Stumble

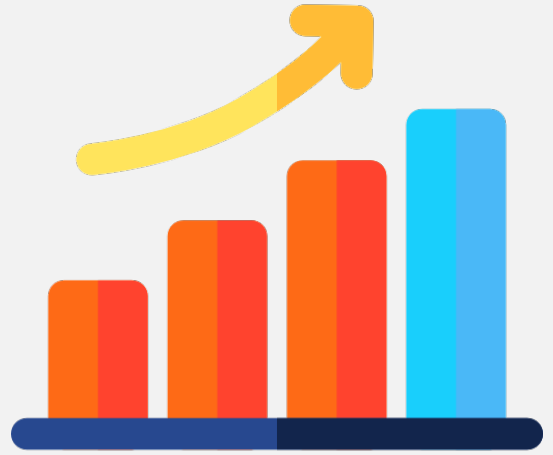
Image source: 非凡新聞

# Introduction to LLMs

## Benefits of using large language models

A single model can be used for different tasks

The fine-tune process requires minimal filed data



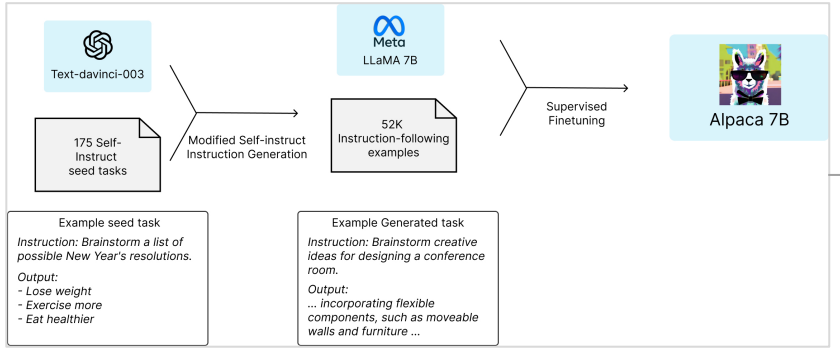
The performance is continuously growing with more data and parameters

Adapted from: Google Cloud
Image source: flaticon

# Introduction to LLMs

## Related LLMs

| Name | BERT | GPT-1~4 | PaLM, PaLM 2 | BLOOM | LLaMA | Alpaca |
|---|---|---|---|---|---|---|
| **Year** | 2018 | 2018 – | Apr. 2022 – | Jul. 2022 – | Feb. 2023 | Mar. 2023 |
| **Developer** | Google | OpenAI | Google | HuggingFace | Meta | Stanford |
| **Params** | 340M | 120M ~ 1T (?) | 540B ~ 1.2T | 175B | 7B ~ 65B | 7B |
| **Corpus size** | 3.3B | 4.5G (≒ 1B) ~ ? | 768B ~ 3.6T | 350B | 1.4T | < 600 USD |

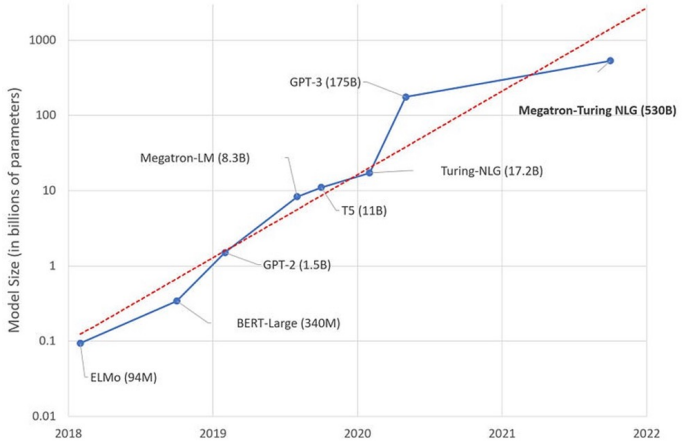



**Academic use only**

Adapted from: Wikipedia -- Large language model

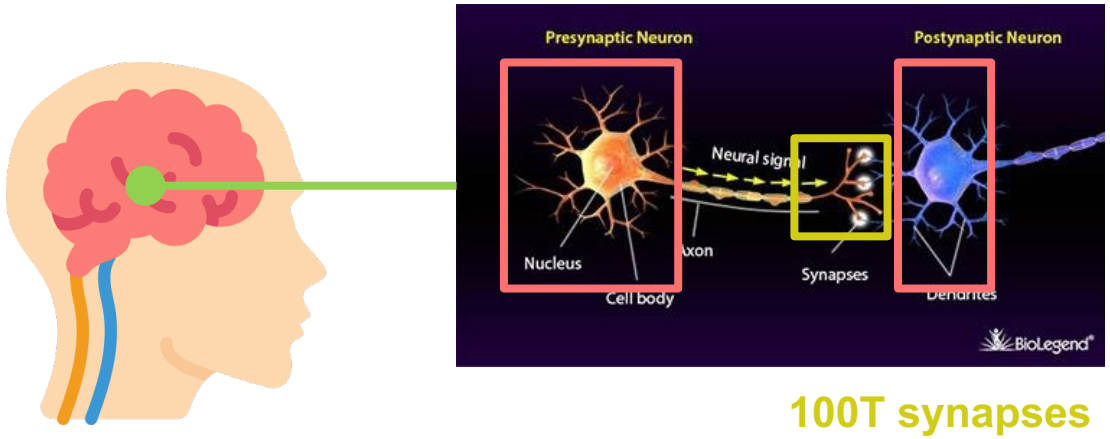
# Introduction to LLMs

## A new Moore's Law?

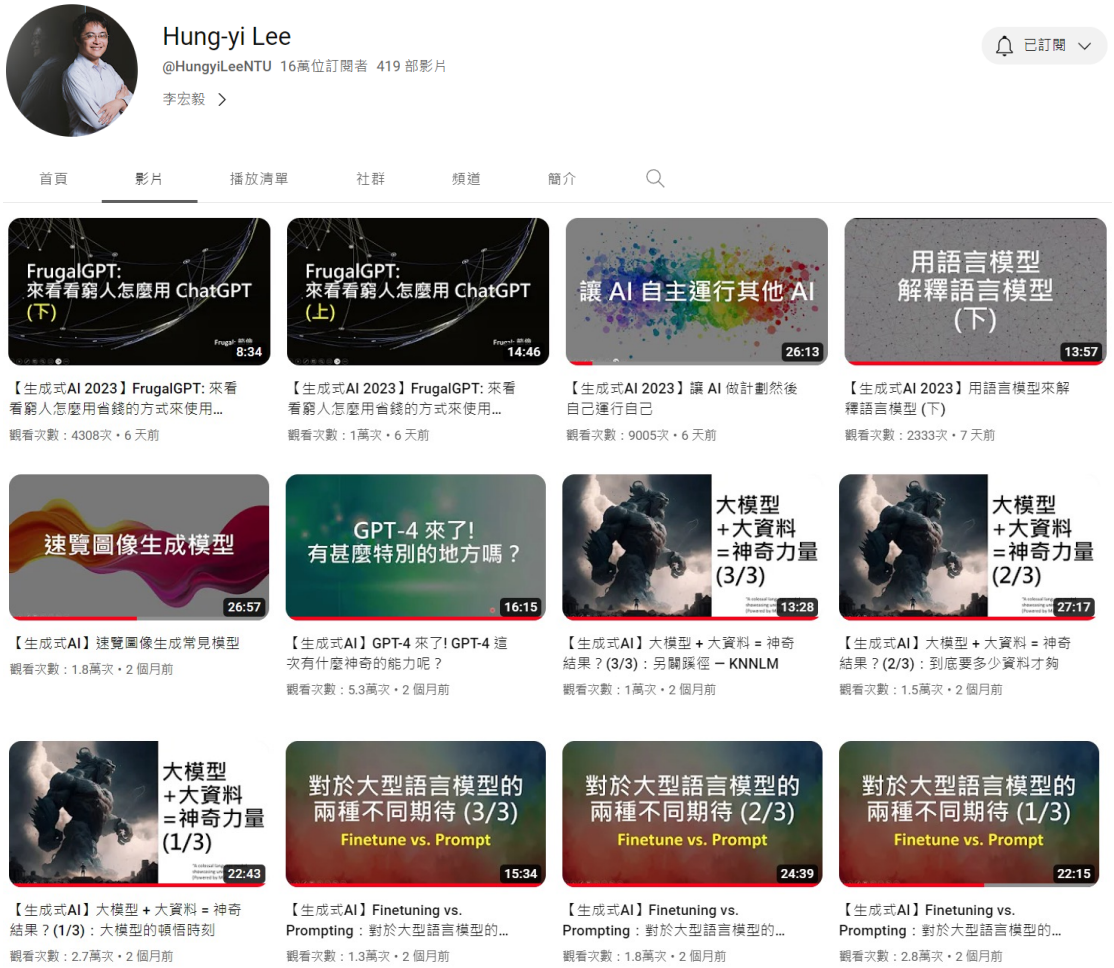



→ ~ 10x / year

86B neurons




100T synapses



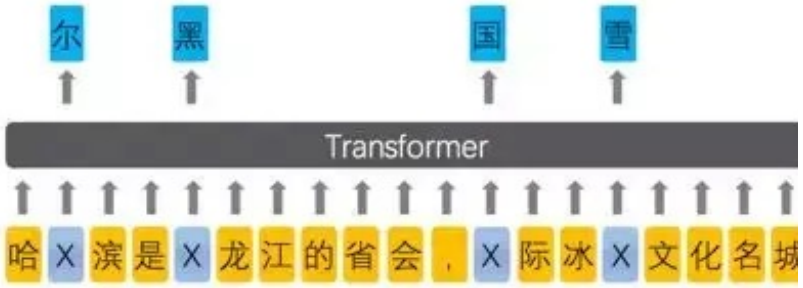

Adapted from: Large Language Models: A New Moore's Law?
Image source: flaticon, BioLegend

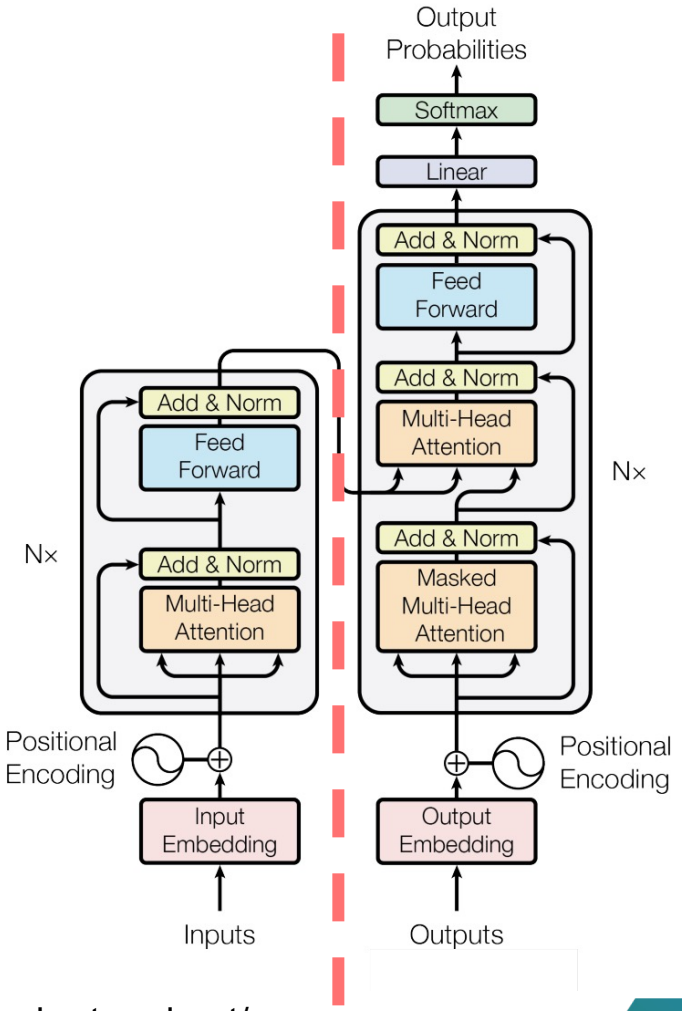
# Introduction to LLMs

## Self-supervised learning or Semi-supervised learning

**BERT**

文字填空




**GPT**

文字接龍

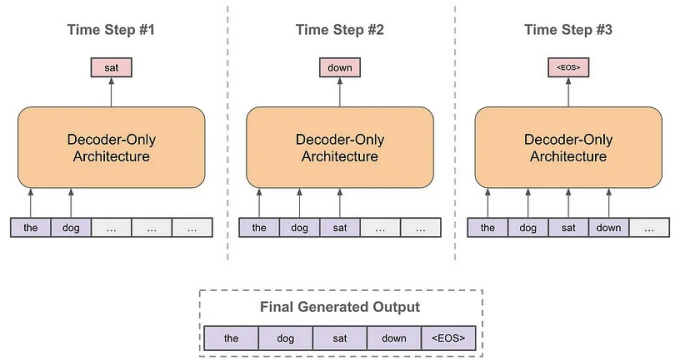
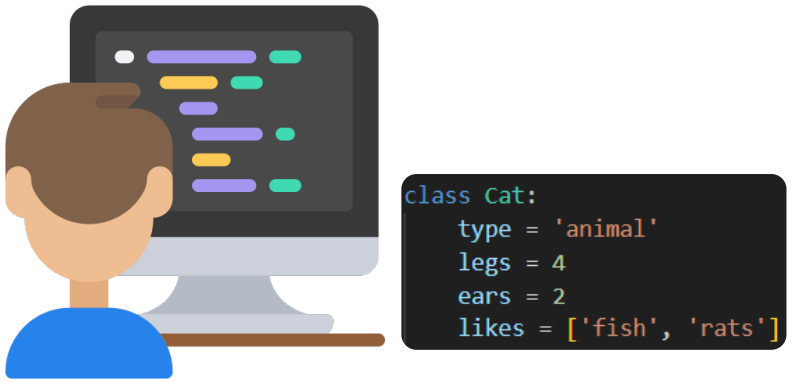



Image source:
https://zhuanlan.zhihu.com/p/59436589
https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/
https://towardsdatascience.com/language-models-gpt-and-gpt-2-8bdb9867c50a

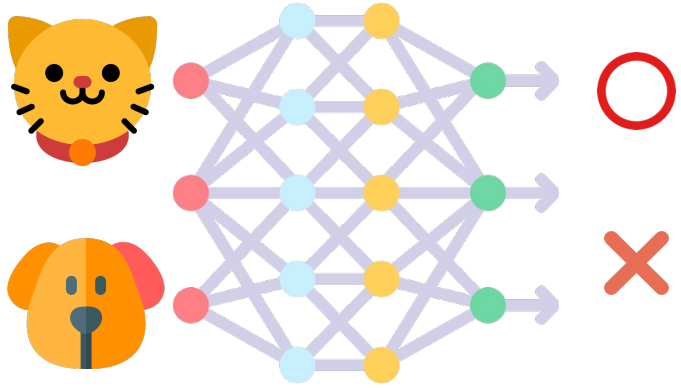
# Introduction to LLMs

## 3 steps of programming

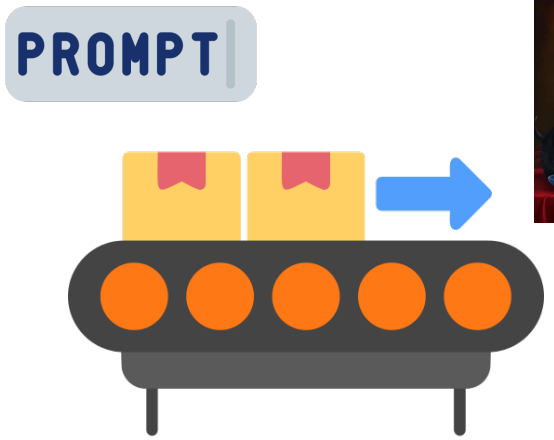
**Traditional**




**Neural Network**



**Generative Language Model**




Adapted from: Google Cloud
Image source: flaticon, 吳淡如FB

## LLM Development vs. Traditional Development

| Traditional ML Development | LLM Development (by pretrained APIs) |
| --- | --- |
| ☑ ML expertise needed | ☒ ML expertise needed |
| ☑ training examples | ☒ training examples |
| ☑ need to train a model | ☒ need to train a model |
| ☑ compute time + hardware | Thinks about prompt design |
| Thinks about minimizing a loss function | |

Adapted from: Google Cloud

Prompt means all of the text that we feed into an LLM, which figures out what text to feed LLM to take on the behavior you want.

Adapted from: Google Cloud

# Introduction to LLMs

## Prompt (提示語)

### Prompt Design

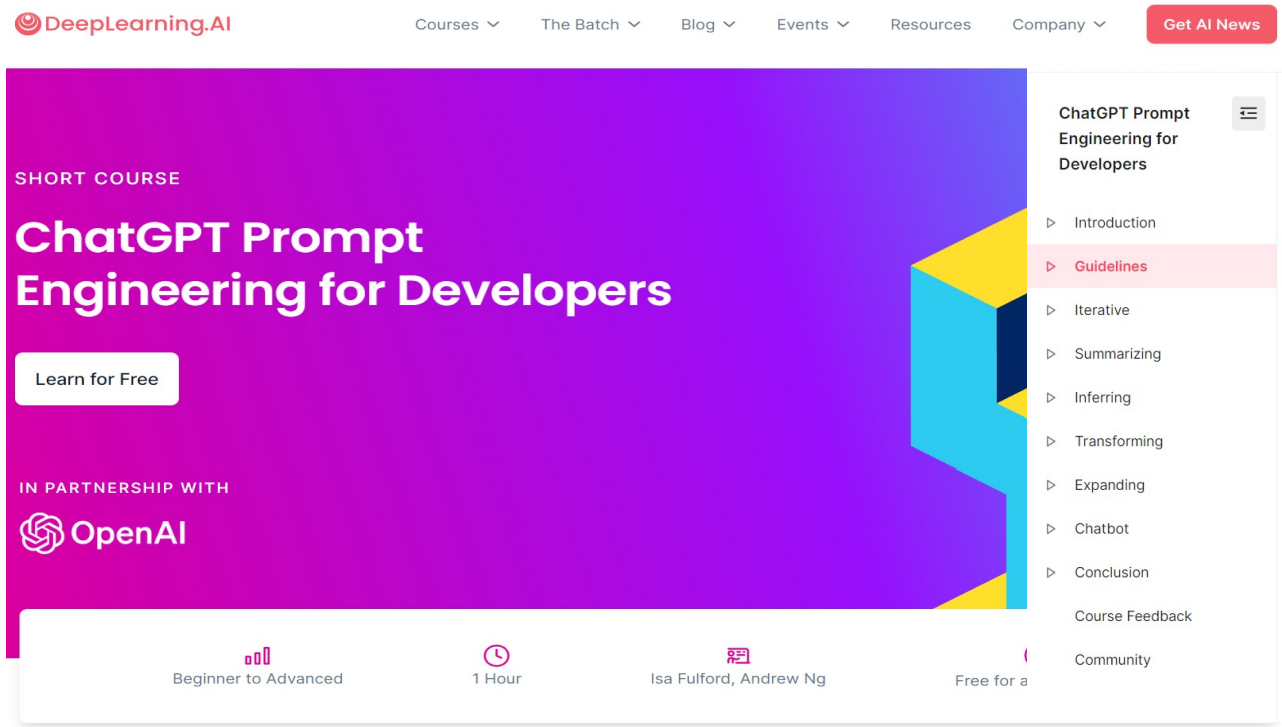
- Prompts involve **instructions and context** passed to a language model to achieve a desired task.


### Prompt Engineering

- Prompt engineering is the practice of **developing and optimizing prompts** to efficiently use language models for a variety of applications. (by **keywords** or **examples**)

Adapted from: Google Cloud, 非凡新聞

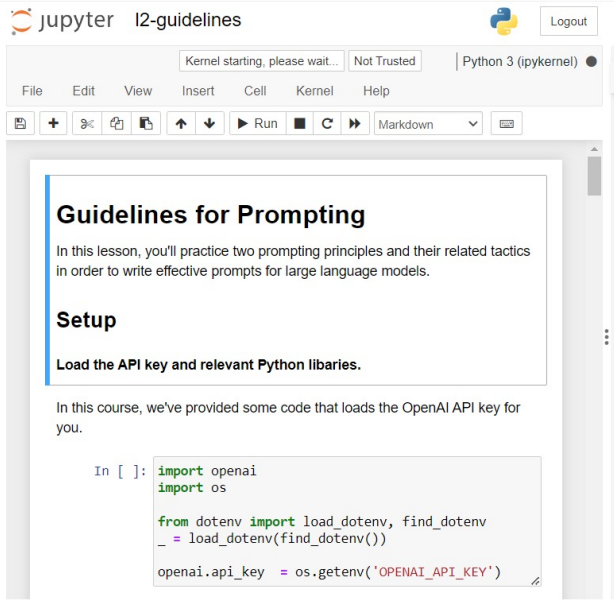
# Prompt Engineering (提示語工程)

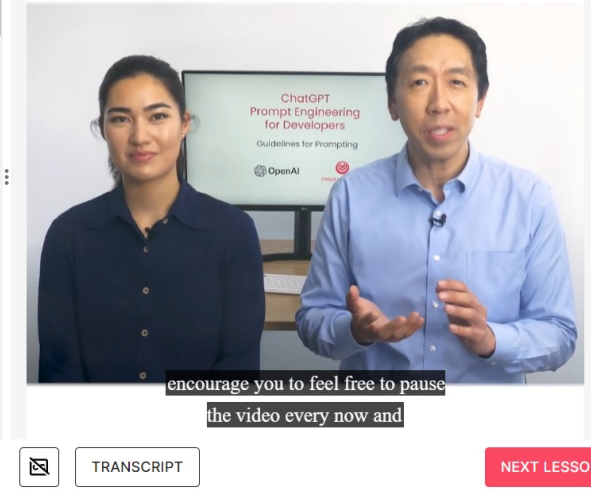



Learn prompt engineering best practices for application development

Discover new ways to use LLMs, including how to build your own custom chatbot

Gain hands-on practice writing and iterating on prompts yourself using the OpenAI API



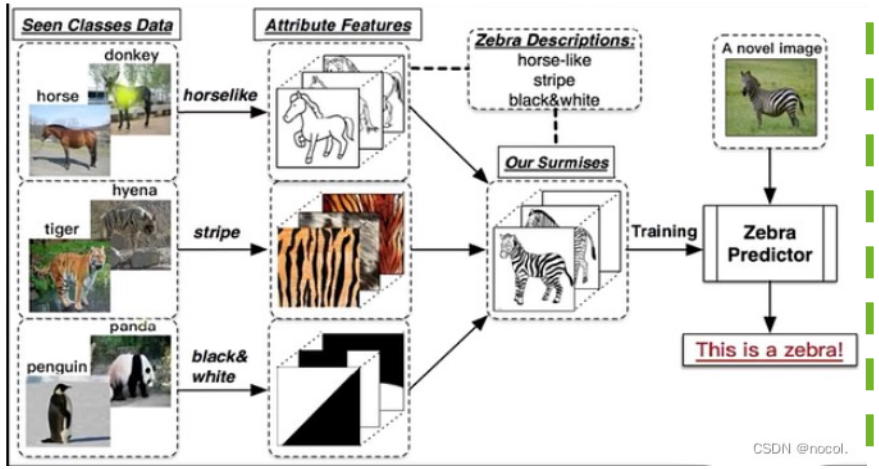

Isa Fulford Andrew Ng

source: DeepLearning.AI

# Zero-shot, One-shot and Few-shot Learning

第一部分：單題 (第1 - 23題，共23題)

1. Look at the picture. The man is holding a _____ of grapes in his hands.
(A) bag
(B) basket
(C) bowl
(D) box



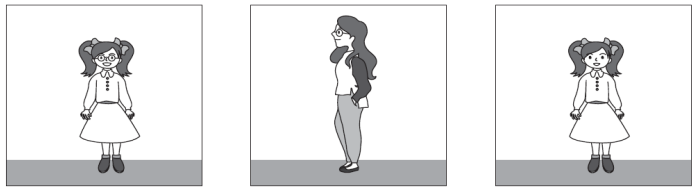

Zero-shot

第一部分：辨識句意 (第1 - 3題)

作答說明：每題均有三張圖片，請依據所聽到的句子，選出符合描述的圖片，每題播放兩次。

示例題：你會看到

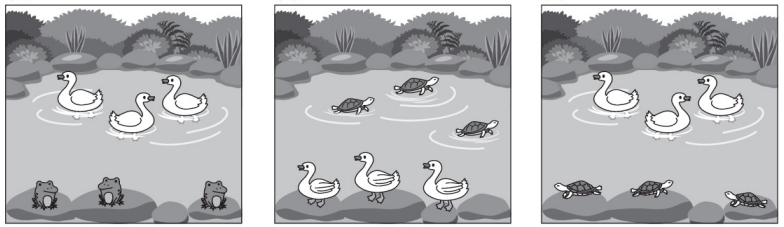
(A) (B) (C)



然後你會聽到……(播音)。依據所播放的內容，正確答案應該選A，請將答案卡該題「Ⓐ」的地方塗黑、塗滿，即：● Ⓑ Ⓒ

第 1 題

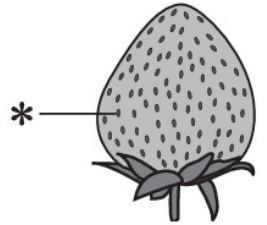
(A) (B) (C)



One-shot

35. 圖(二十)為草莓花朵構造及其發育的示意圖，已知草莓是由花托處膨大而來，若圖(二十一)中的＊構造是由草莓的子房發育而成，則此＊構造應稱為下列何者？


圖(二十)




圖(二十一)

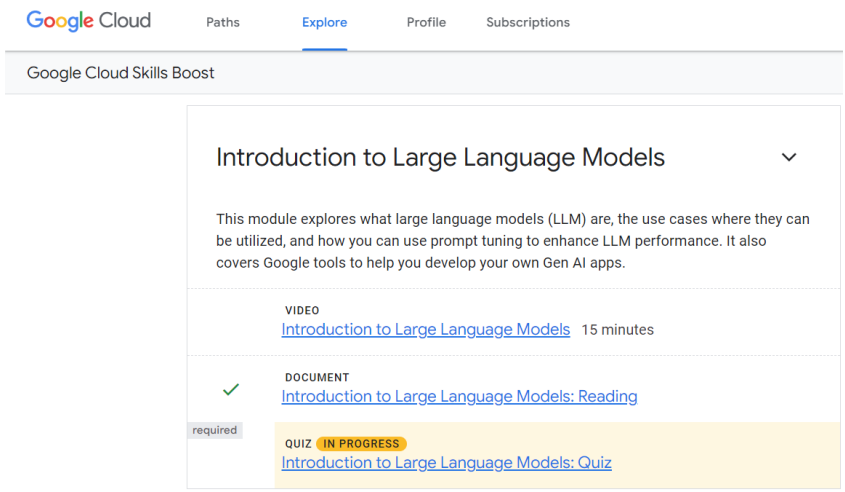
(A)胚珠 (B)種子 (C)果實 (D)花粉

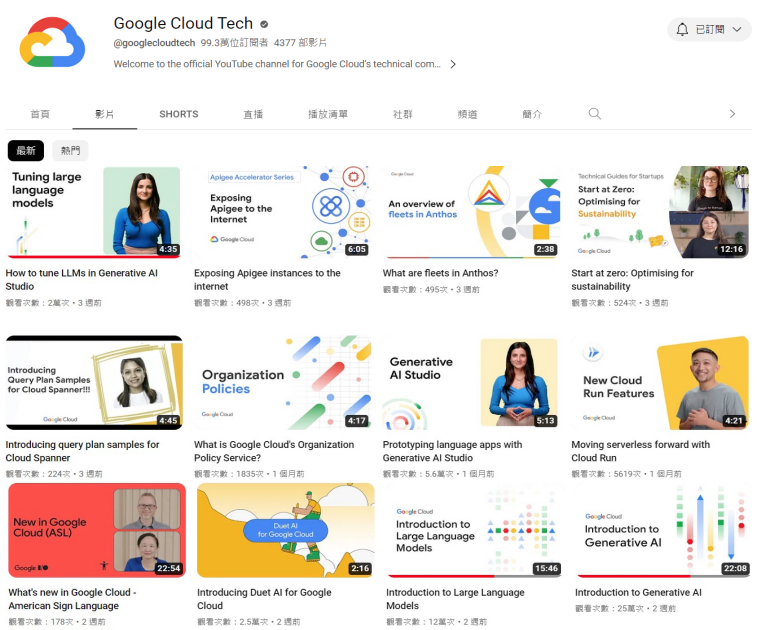
Few-shot

Image source:
國立臺灣師範大學心理與教育測驗研究發展中心
https://blog.csdn.net/zcyzcyjava/article/details/127006287

# Introduction to LLMs

## For more details, please refer to Google Cloud …



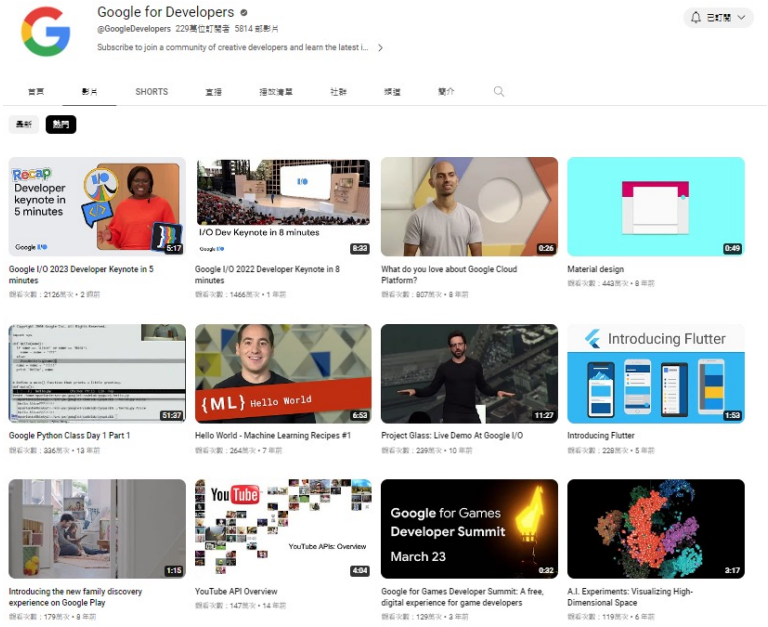
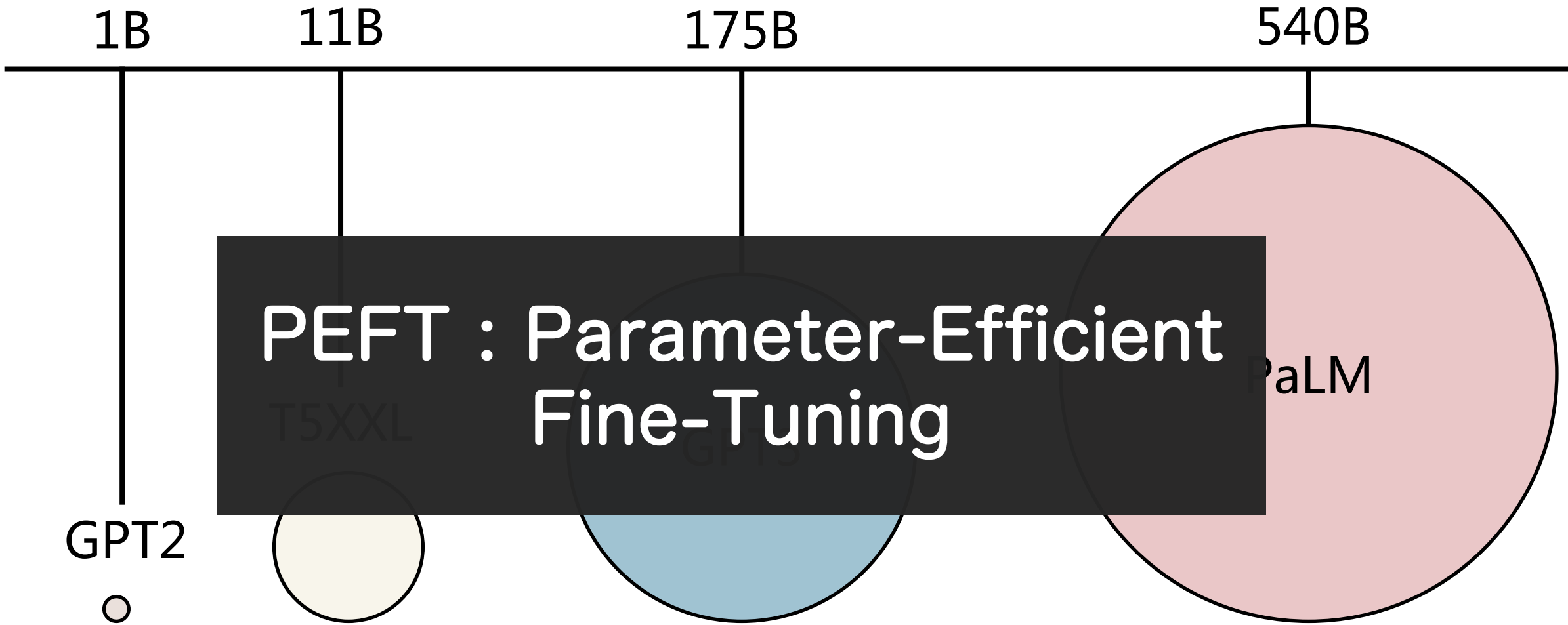




Image source: Google Cloud

# The problem with finetuneing LLM



# Parameter-Efficient Fine-Tuning

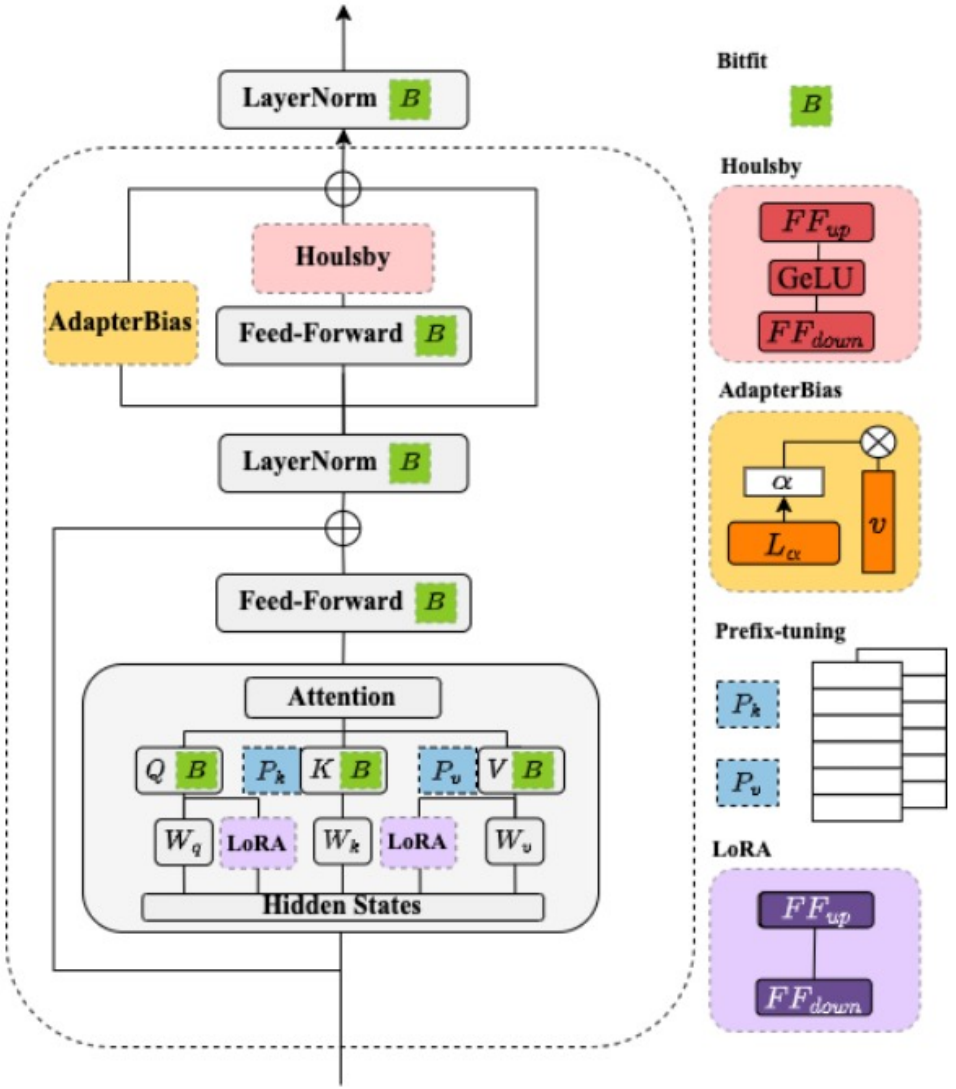
- **Adapter**：透過添加額外的模型架構，並固定 Freeze LLM 的模型參數，進行訓練。
- **Prefixing**：Prefixing 在 Prompt 的前半部添加 Token 來讓模型對於特定任務可以做得更好。

# Parameter-Efficient Fine-Tuning

- **LoRA** : LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS
- **Prefix Tuning** : Prefix-Tuning: Optimizing Continuous Prompts for Generation, P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks
- **P-Tuning** : GPT Understands, Too
- **Prompt Tuning** : The Power of Scale for Parameter-Efficient Prompt Tuning

# Adapter

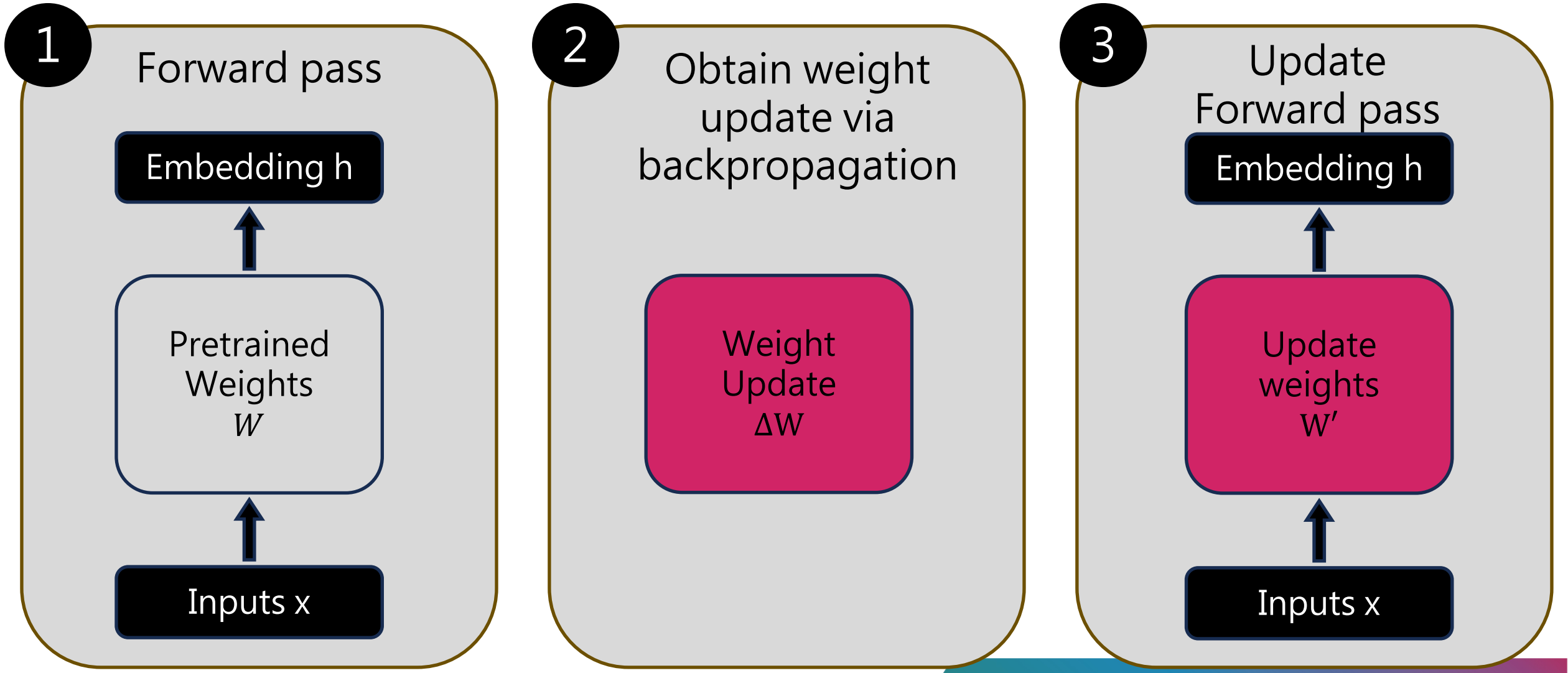
Adapter-hub



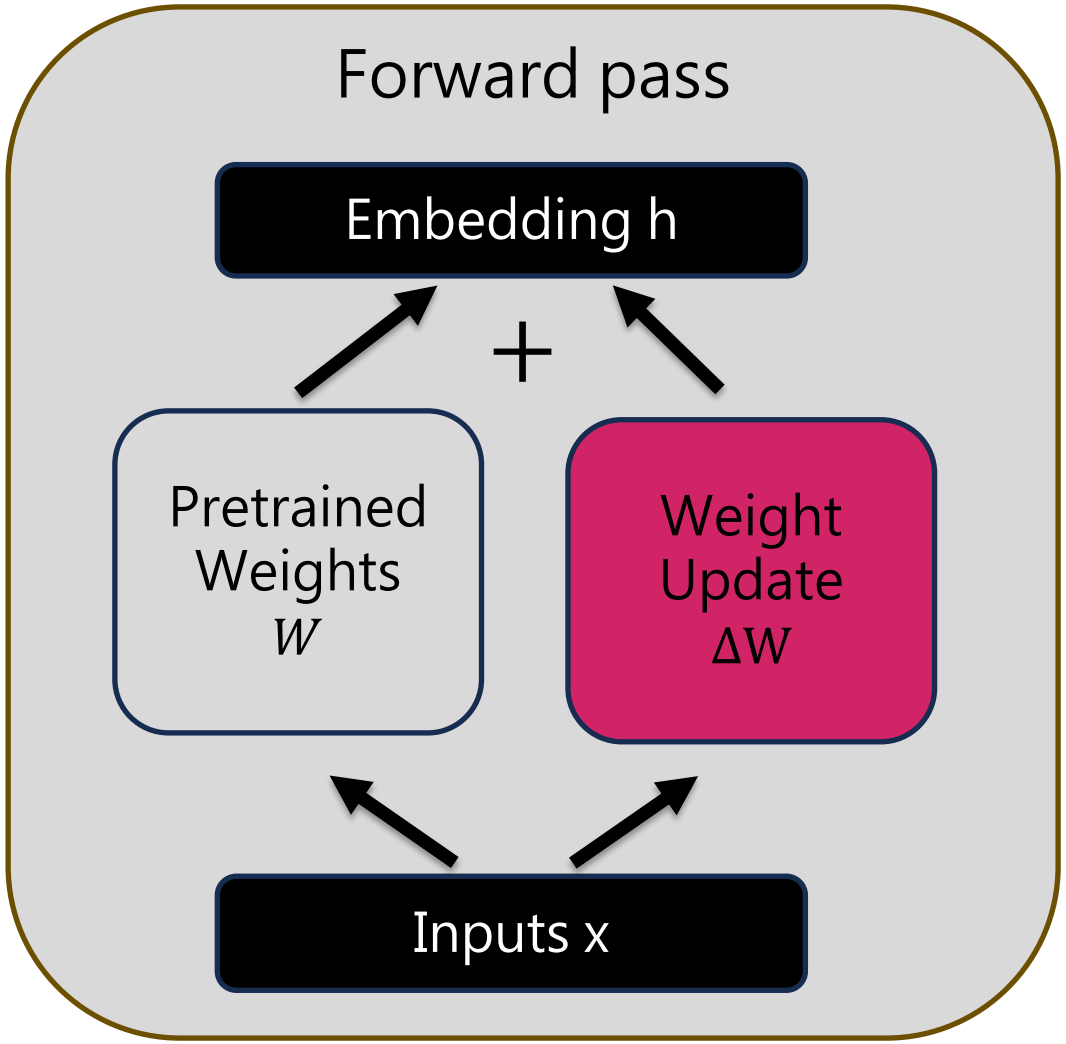

【生成式AI】Finetuning vs. Prompting：
對於大型語言模型的不同期待所衍生的兩類使用方式 (1/3)

EXPLORING EFFICIENT-TUNING METHODS IN SELF-SUPERVISED SPEECH MODELS

# LoRA : Low Rank Adaptation

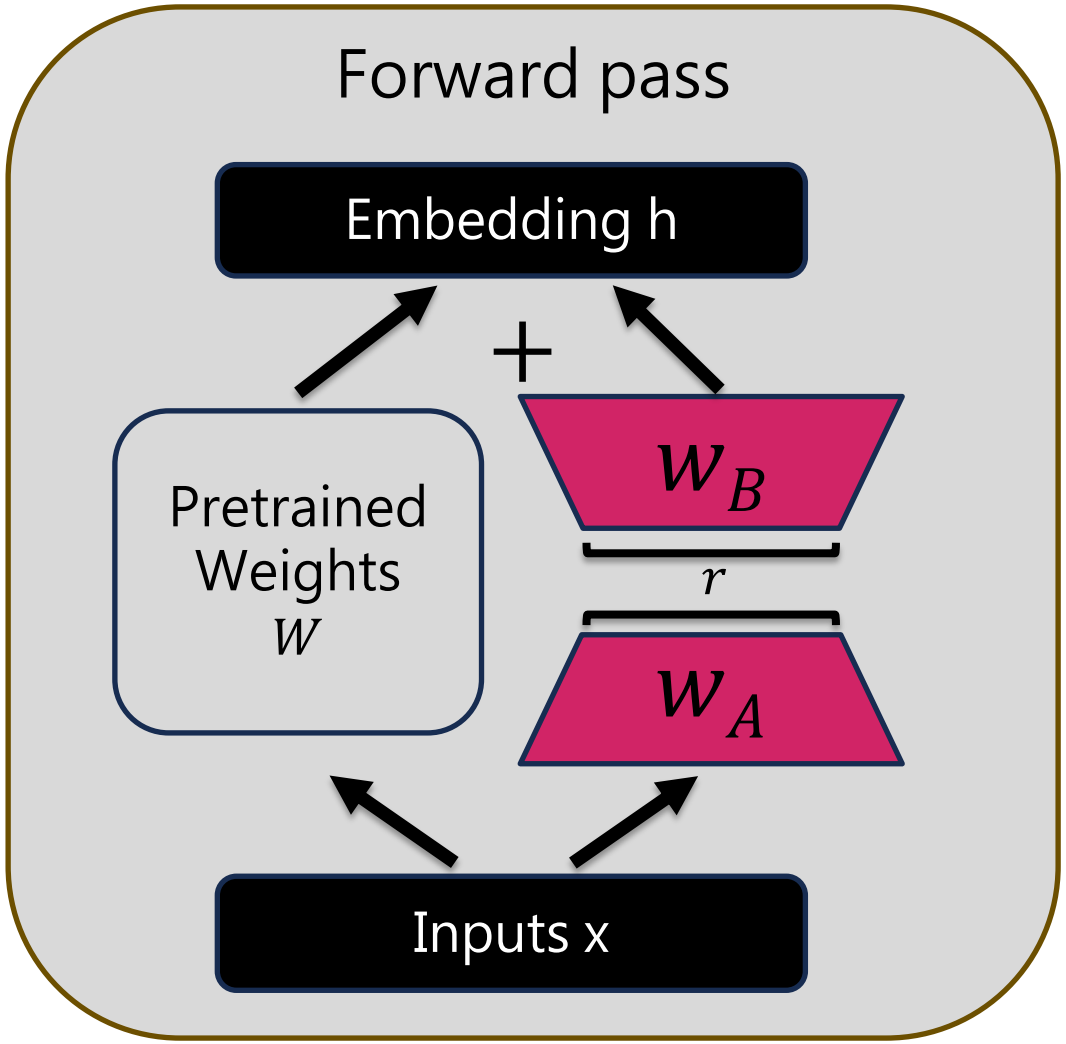


# LoRA : Low Rank Adaptation



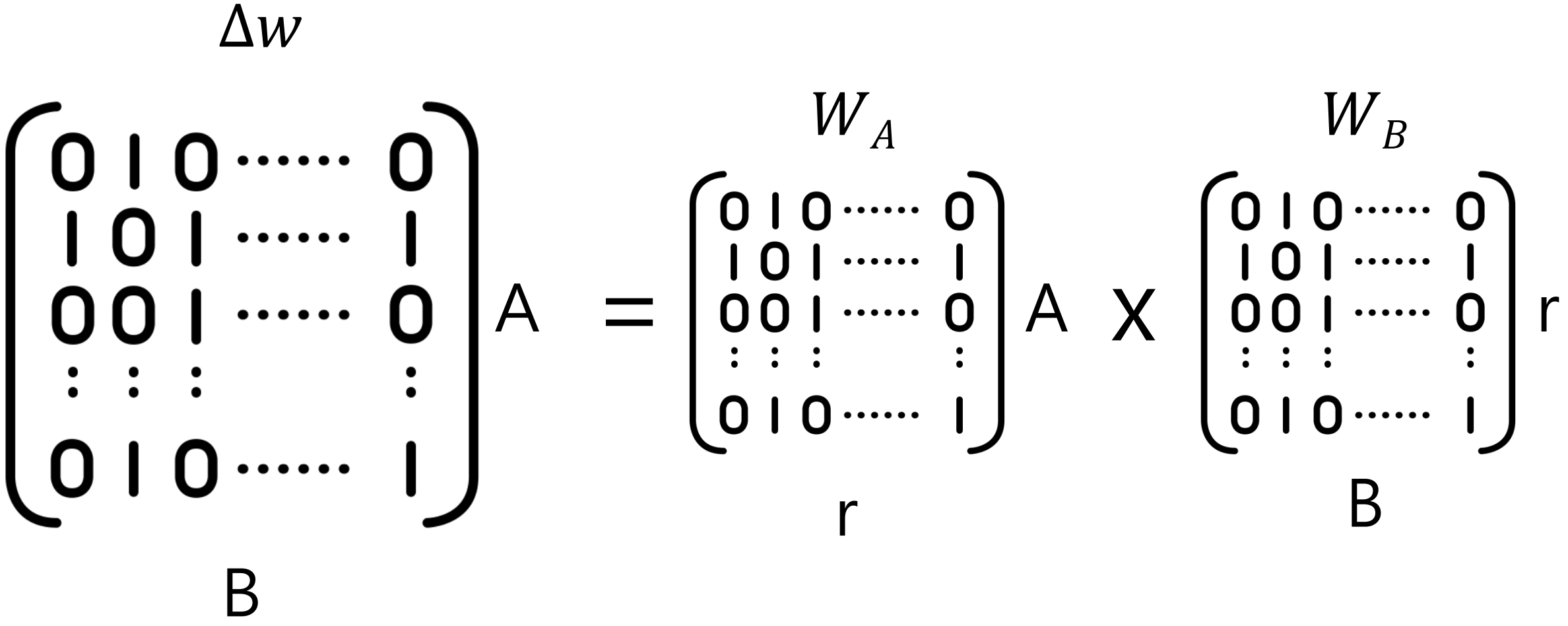

$$h = wx + \Delta wx$$

# LoRA : Low Rank Adaptation



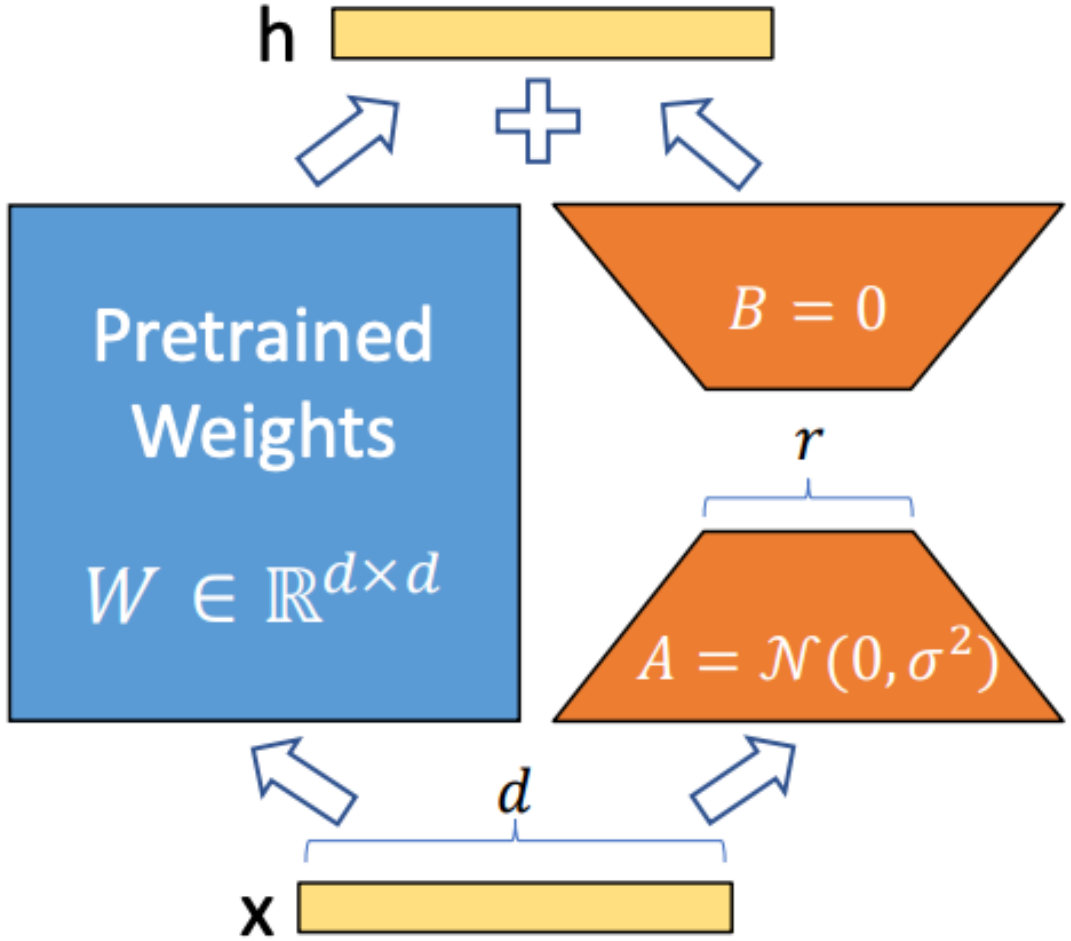

$$h = wx + w_A w_B$$

# LoRA explain



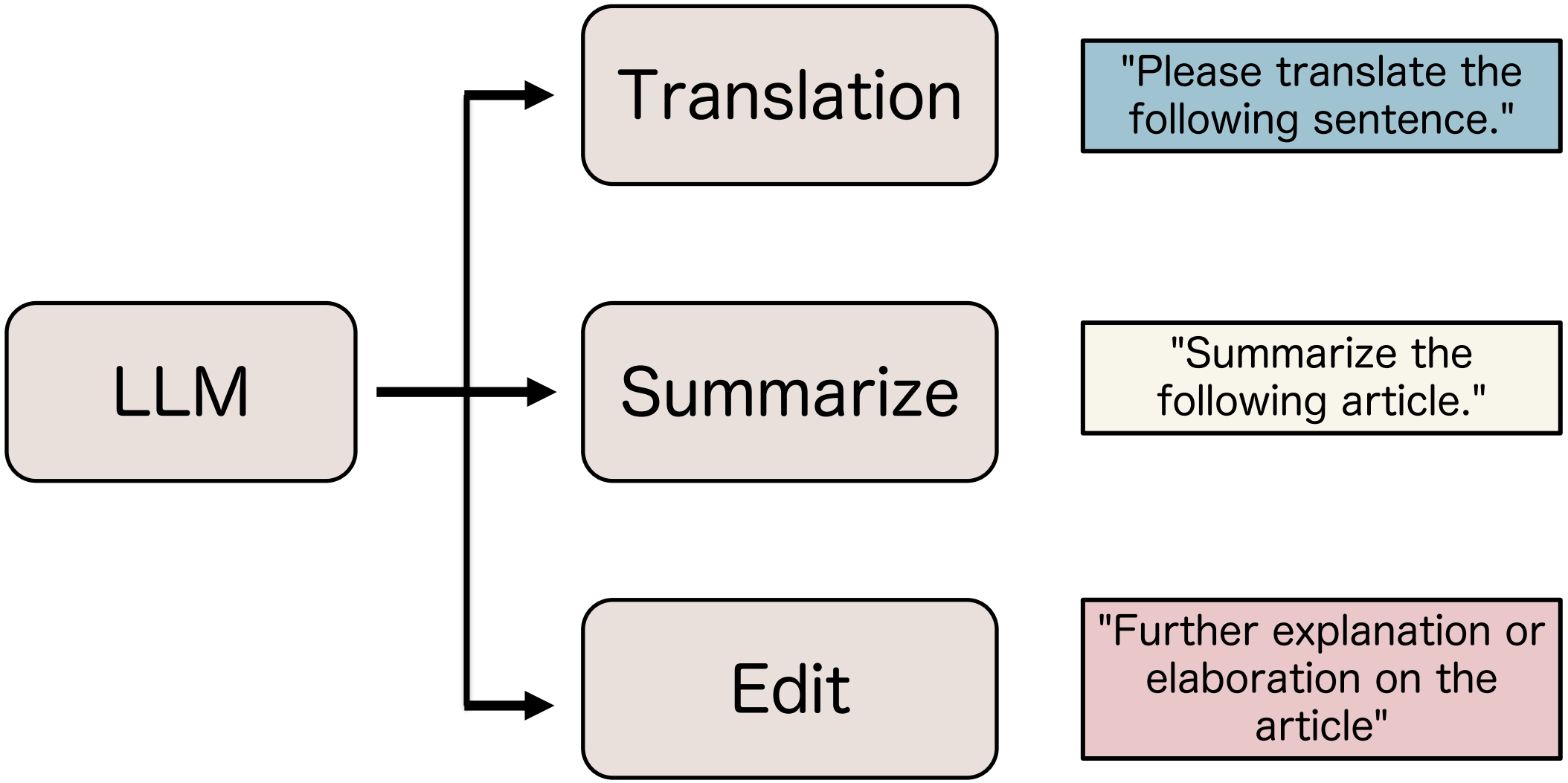
# LoRA：Low Rank Adaptation

Low-Rank Adaptation (LoRA)，概念是透過凍結原本的預訓練模型(e.g., GPT-3) 的權重，搭配一個小的模型進行微調就可以達到很好的 Fine-Tuning 效果




LoRA: Low-Rank Adaptation of Large Language Models

# Prompt




【生成式AI】Finetuning vs. Prompting：
對於大型語言模型的不同期待所衍生的兩類使用方式 (1/3)

# Discrete/hard prompts

## Discrete/hard prompts

- natural language instructions/task descriptions

Good morning

## Problems

- requiring domain expertise/understanding of the model's inner workings
- performance still lags far behind SotA model tuning results
- sub-optimal and sensitive

請 翻 譯 ：早 安

# Sub-optimal and sensitive hard prompts

Difficulty of manually designing prompts

1. Prompts that humans consider reasonable is not necessarily effective for language models(Liu et al, 2021)
2. Pre-trained LMs are sensitive to the choice of prompts (Zhao et al., 2021)

| Prompt | P@1 |
|---|---|
| [X] is located in [Y]. *(original)* | 31.29 |
| [X] is located in which country or state? [Y]. | 19.78 |
| [X] is located in which country? [Y]. | 31.40 |
| [X] is located in which country? In [Y]. | 51.08 |

# Shifting from hard to soft prompts

**Progress in prompt-based learning**

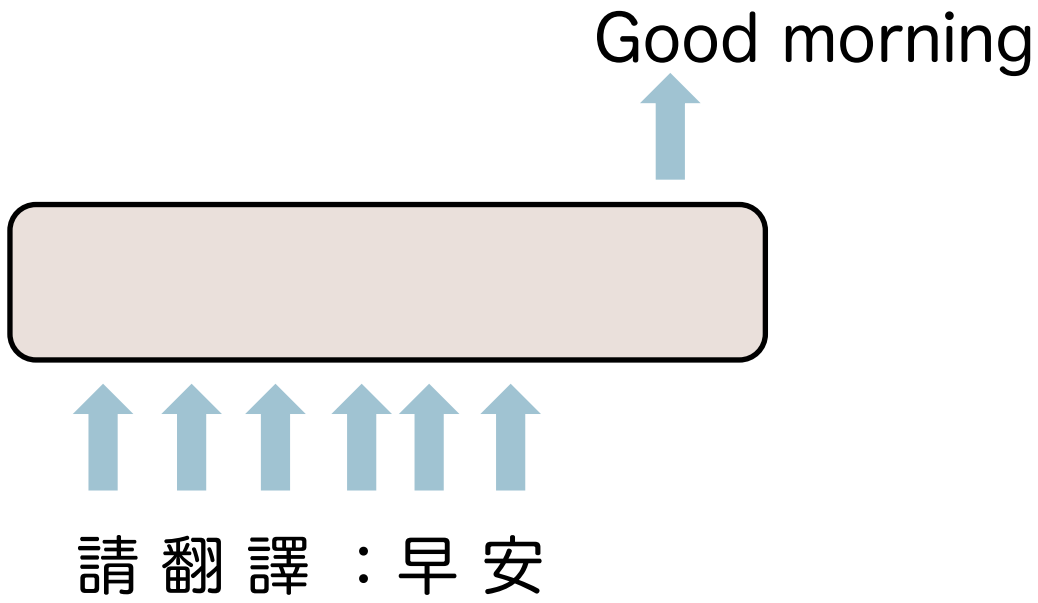
- manual prompt design (Brown et al., 2020; Schick and Schutze, 2021a,b)
- mining and paraphrasing based methods to automatically augment the prompt sets (Jiang et al., 2020)
- gradient-based search for improved discrete/hard prompts (Shin et al., 2020)
- automatic prompt generation using a separate generative language model (i.e., T5) (Gao et al., 2020)
- learning continuous/soft prompts (Liu et al., 2021; Li and Liang., 2021; Qin and Eisner., 2021; Lester et al., 2021)

**Continuous/soft prompts**

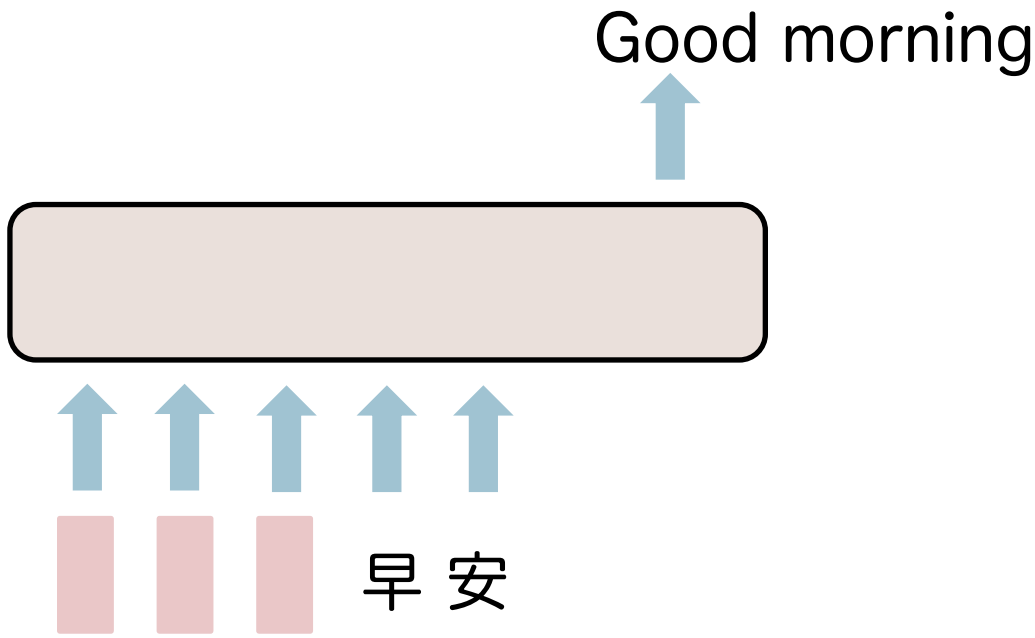
- additional learnable parameters injected into the model

# Soft prompt

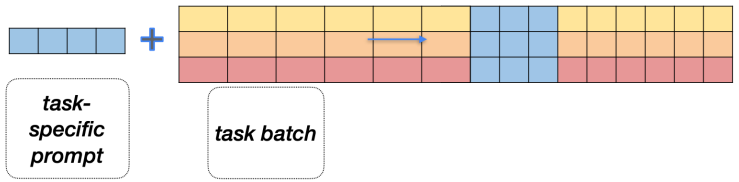
## Discrete/hard prompts



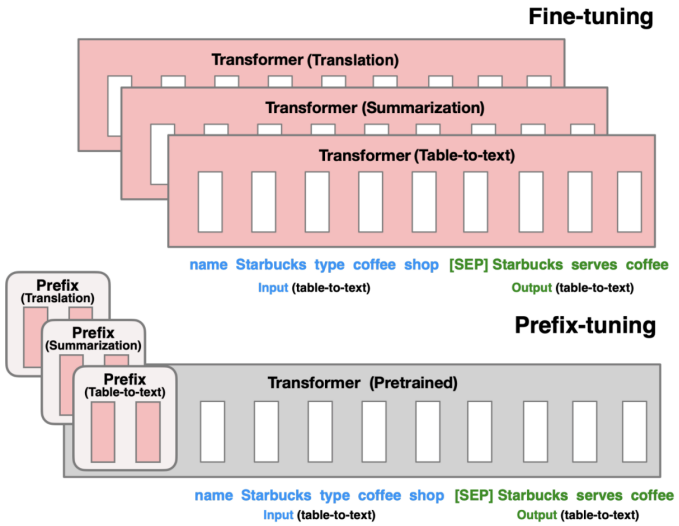

## Continuous/soft prompts



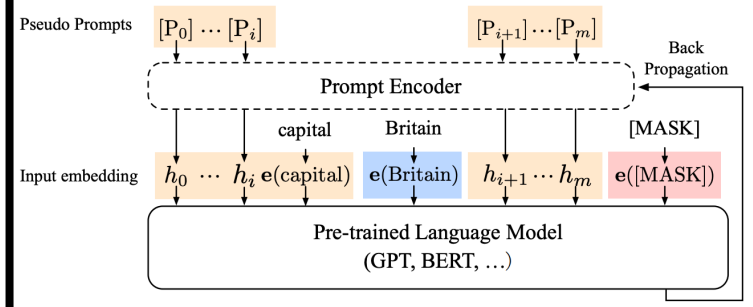
# Learn soft prompts effectively




Prompt tuning

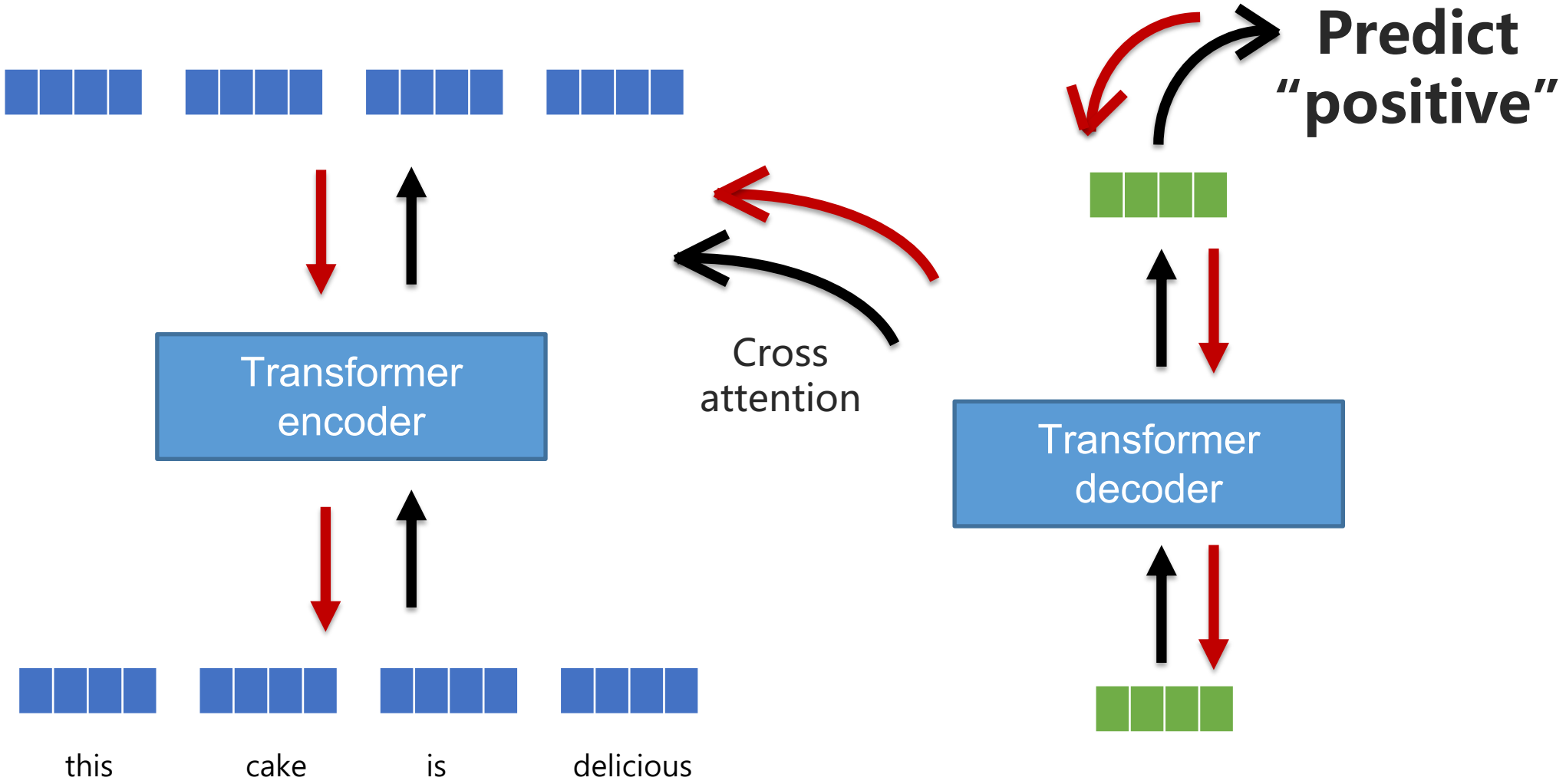



Prefix tuning



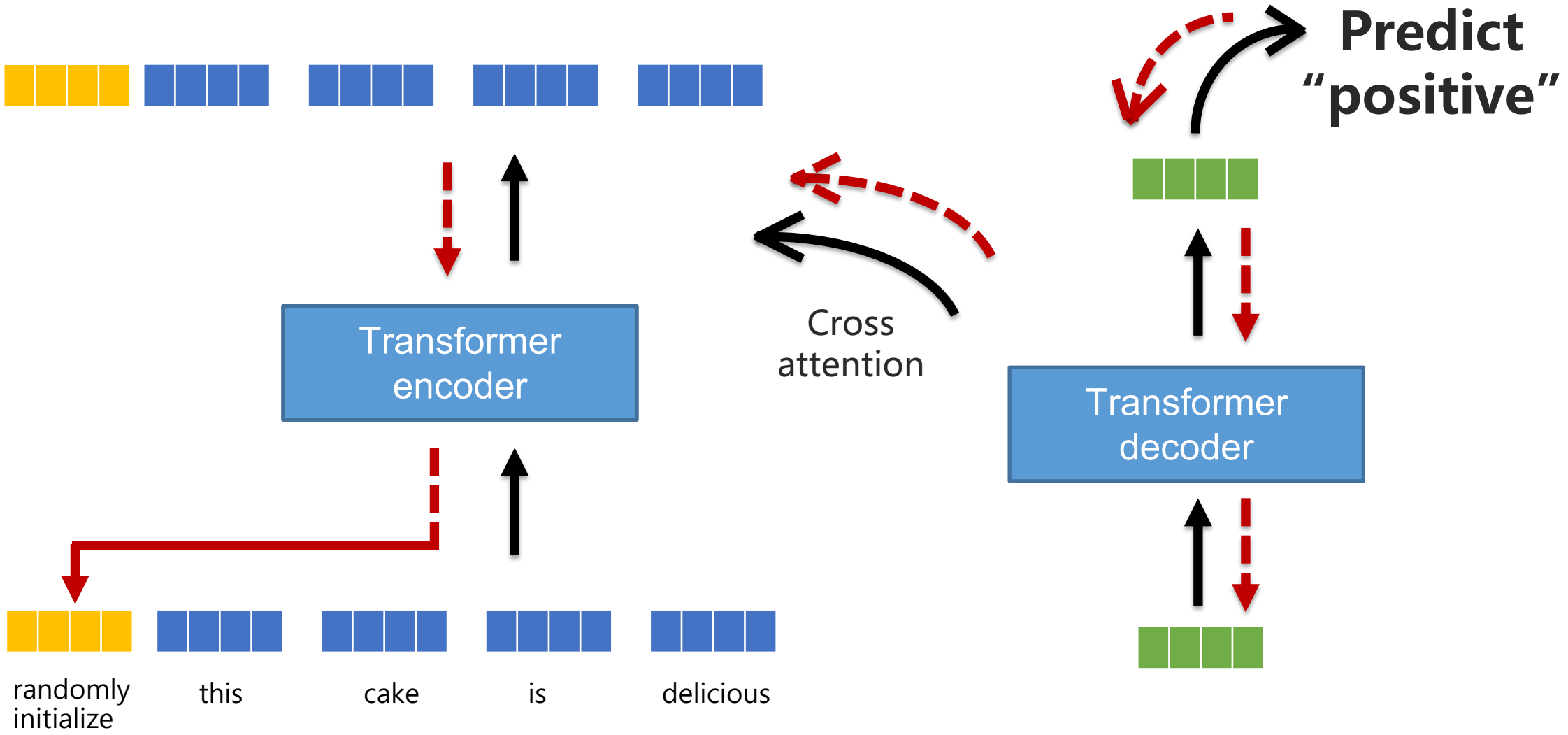

P tuning

# Prompt tunning



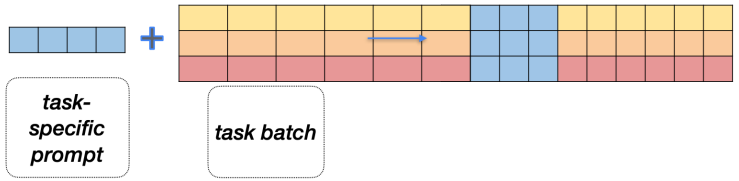

Lester et al. 2021

# Prompt tunning

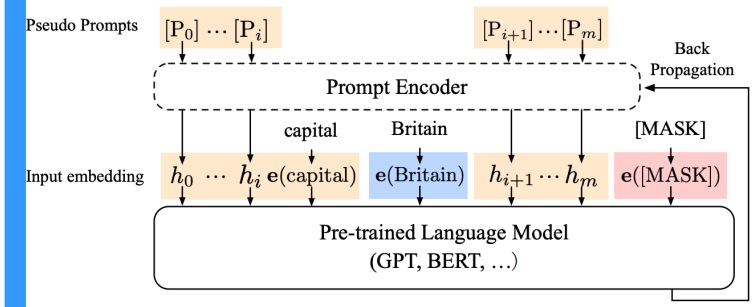



Lester et al. 2021

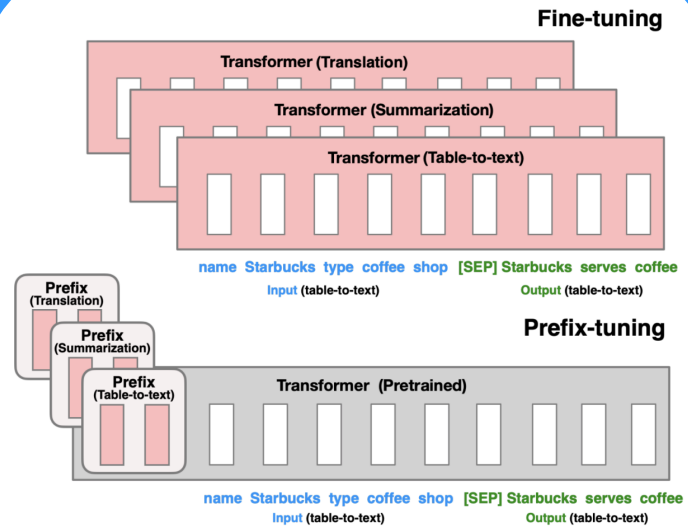
# Learn soft prompts effectively




Prompt tuning



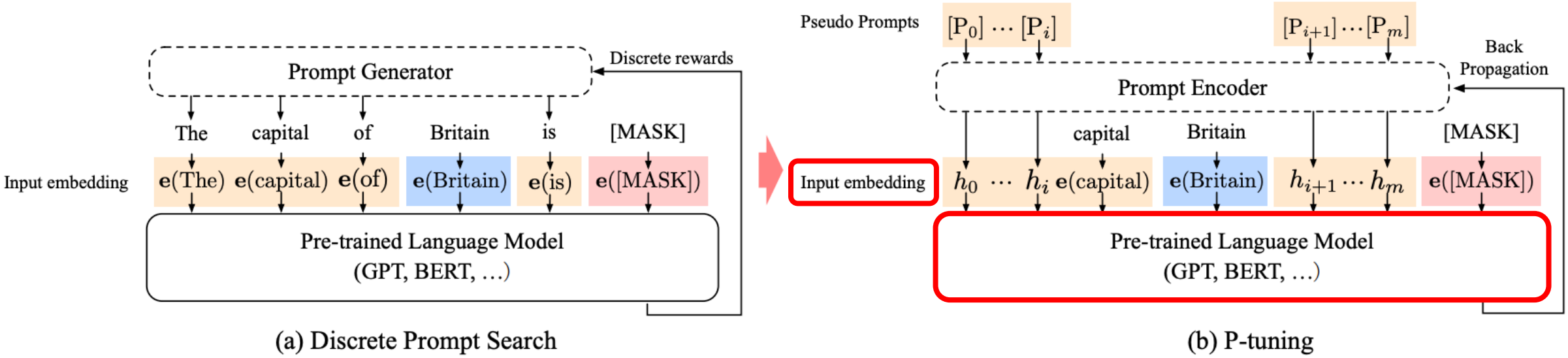

P tuning




Prefix tuning

# P-Tuning

Direct optimize the embeddings instead of prompt tokens.




(a) Discrete Prompt Search

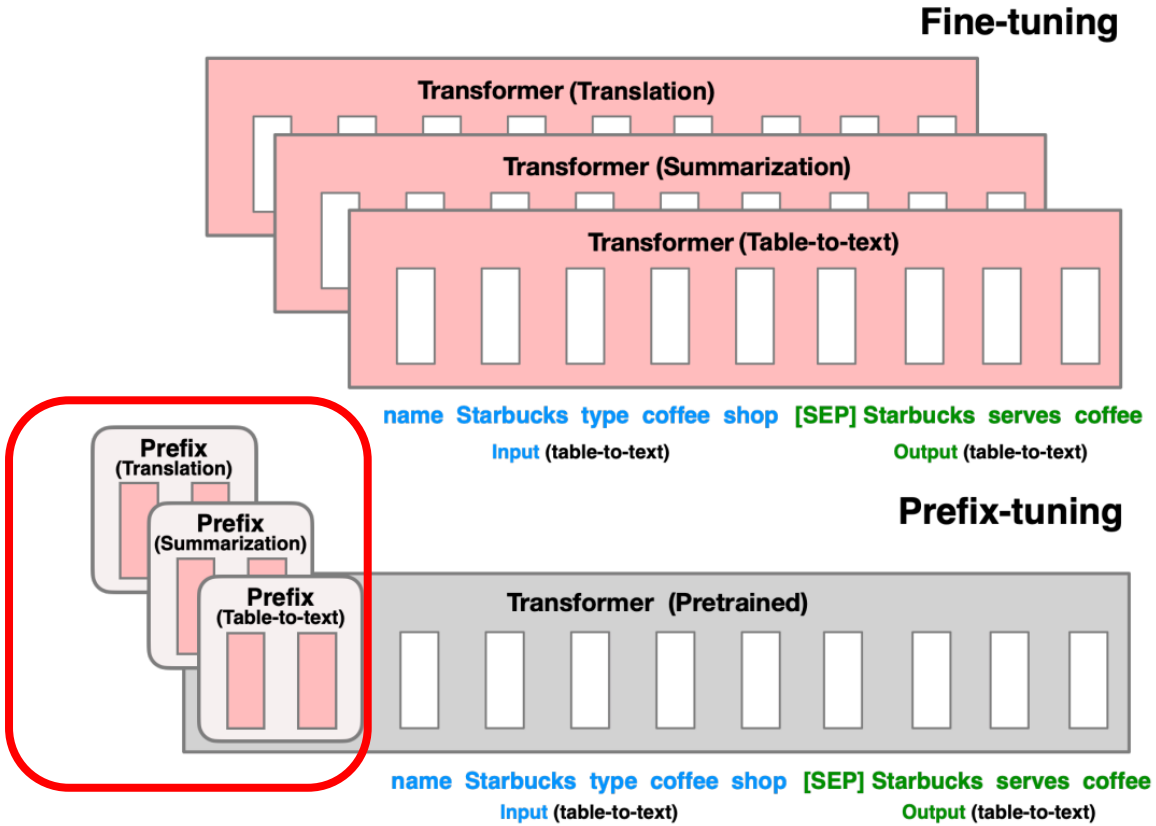
(b) P-tuning

| | Prompt | $\mathcal{D}_{dev}$ Acc. | $\mathcal{D}_{dev32}$ Acc. |
|---|---|---|---|
| **Hard Prompt** | Does [PRE] agree with [HYP]? [MASK]. | 57.16 | 53.12 |
| | Does [HYP] agree with [PRE]? [MASK]. | 51.38 | 50.00 |
| | Premise: [PRE] Hypothesis: [HYP] Answer: [MASK]. | 68.59 | 55.20 |
| | [PRE] question: [HYP]. true or false? answer: [MASK]. | 70.15 | 53.12 |
| **Soft Prompt** | P-tuning | 76.45 | 56.25 |

https://arxiv.org/pdf/2103.10385.pdf

# Prefix-Tuning

- Only optimize the prefix embeddings(all layers) for efficiency
- Prefix tuning stores 1000x fewer parameters than a fully finetuned model




**Efficiency**

**for time and space**

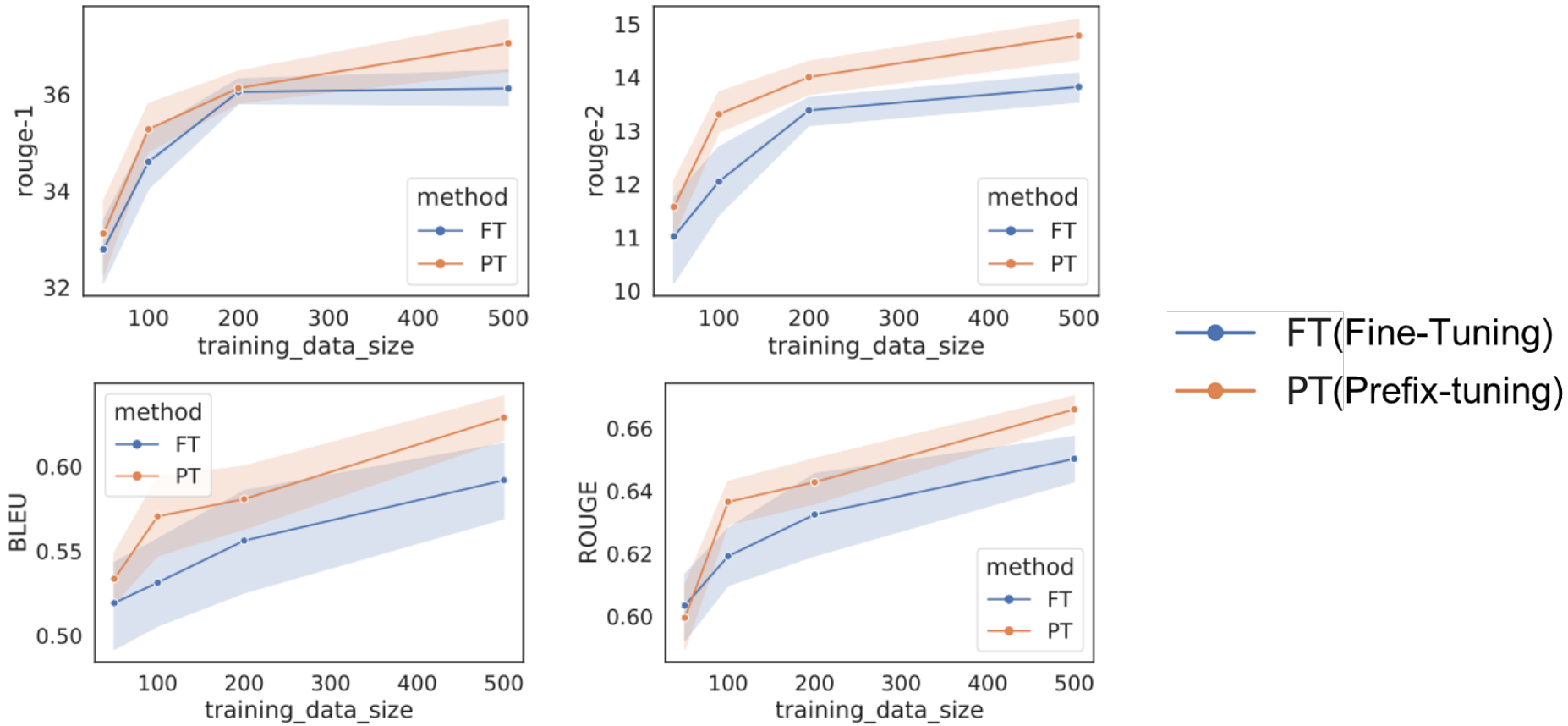
https://arxiv.org/pdf/2101.00190.pdf

# Prefix-Tuning

- PreÞx-tuning has a comparative advantage when the number of **training examples is smaller**.

**Summarization**

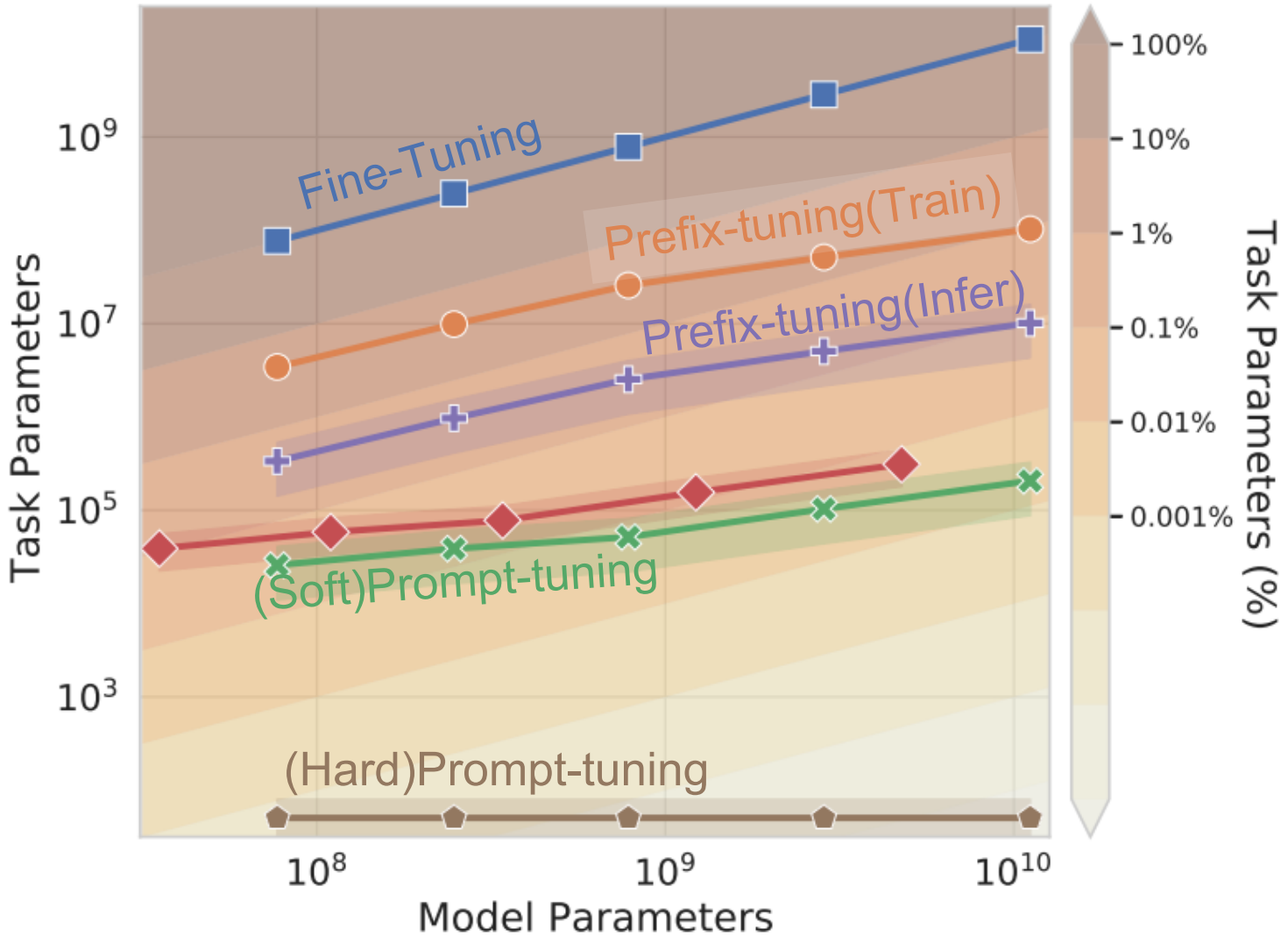
**Table-to-text**




- Y-axis is the evaluation metric (higher is better)

https://arxiv.org/pdf/2101.00190.pdf

# Prefix-Tuning, Prompt-Tuning




https://arxiv.org/pdf/2104.08691.pdf

# Applied Deep Learning 陳縕儂老師影片

- 台大資訊 深度學習之應用 | ADL 15.1: Issues of PLMs 如何提示預訓練模型
- 台大資訊 深度學習之應用 | ADL 15.2: (Hard) Prompt-Tuning, LM-BFF 用自然語言提示模型
- 台大資訊 深度學習之應用 | ADL 15.3: (Soft) Prompt-Tuning (P-Tuning, Prefix Tuning) 人不懂沒關係機器懂就好
- 台大資訊 深度學習之應用 | ADL 15.4: Instruction Tuning 讓機器了解任務指令
- 台大資訊 深度學習之應用 | ADL 15.5: Prompting Paradigm 基於提示的研究大補帖

# AACL 2022 Tutorial

- Recent Advances in Pre-trained Language Models: Why Do They Work and How to Use Them
- **https://d223302.github.io/AACL2022-Pretrain-Language-Model-Tutorial/**

# Relative papers for Tuning

- **REVIEW Paper:**
  - **Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, https://arxiv.org/pdf/2107.13586.pdf**
- **P-Tuning:**
  - **GPT Understands, Too, https://arxiv.org/pdf/2103.10385.pdf**
  - **P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks, https://arxiv.org/pdf/2110.07602.pdf**
- **Prompt Tuning:**
  - **The Power of Scale for Parameter-Efficient Prompt Tuning, https://arxiv.org/pdf/2104.08691.pdf**
- **Prefix-Tuning:**
  - **Prefix-Tuning: Optimizing Continuous Prompts for Generation, https://arxiv.org/pdf/2101.00190.pdf**
- **Soft-prompt:**
  - **Learning How to Ask: Querying LMs with Mixtures of Soft Prompts, https://arxiv.org/pdf/2104.06599.pdf**

# Case Study – OpenAI

## Some pre-departure reminder

**DO NOT use business / private data!**

**When want to train, then create an account ($18 for free).**

### 1. Personal information we collect

We collect information that alone or in combination with other information in our possession could be used to identify you ("Personal Information") as follows:

**Personal Information You Provide**: We may collect Personal Information if you create an account to use our Services or communicate with us as follows:

- *Account Information:* When you create an account with us, we will collect information associated with your account, including your name, contact information, account credentials, payment card information, and transaction history, (collectively, "Account Information").
- *User Content:* When you use our Services, we may collect Personal Information that is included in the input, file uploads, or feedback that you provide to our Services ("Content").

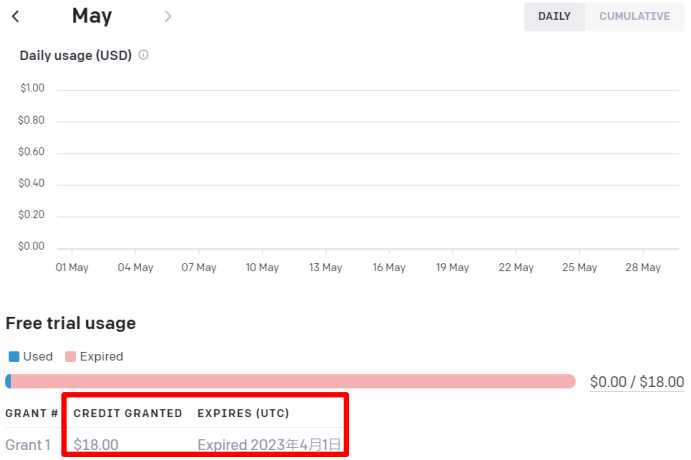



Image source: flaticon

# Case Study – OpenAI

## Work flow

### Step 4: Select model

| Model | Training | Usage |
|---|---|---|
| **A**da | $0.0004 / 1K tokens | $0.0016 / 1K tokens |
| **B**abbage | $0.0006 / 1K tokens | $0.0024 / 1K tokens |
| **C**urie | $0.0030 / 1K tokens | $0.0120 / 1K tokens |
| **D**avinci | $0.0300 / 1K tokens | $0.1200 / 1K tokens |

**Step 5: clean data** (delete repeat, split validation set … )

```
openai tools fine_tunes.prepare_data -f <LOCAL_FILE>
```

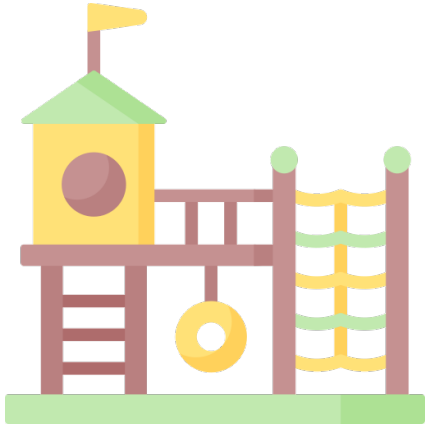
**Step 6: train**

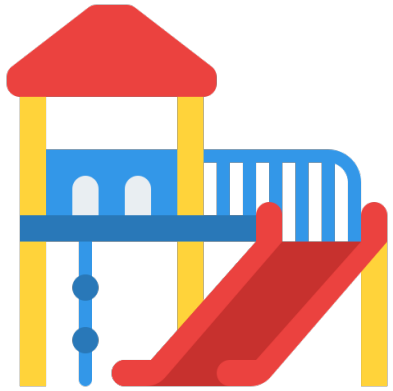
```
openai api fine_tunes.create -t <TRAIN_FILE_ID_OR_PATH> -m <BASE_MODEL>
```

**Step 7: trace the progress**

```
openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>
```

# Case Study – OpenAI





**Play Time**

Image source: flaticon

# Case Study – OpenAI

**Step 7: trace the progress (Garbage message or NOT): 3091 pairs**

```
openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>
```

```
[2023-06-02 16:46:49] Created fine-tune: ft-9xwGZYXgl65dXeW2tLTwYWYH
[2023-06-02 16:49:44] Fine-tune costs $35.15
[2023-06-02 16:49:45] Fine-tune enqueued. Queue number: 3
[2023-06-02 16:51:25] Fine-tune is in the queue. Queue number: 2
[2023-06-02 16:51:49] Fine-tune is in the queue. Queue number: 1
[2023-06-02 16:52:24] Fine-tune is in the queue. Queue number: 0
[2023-06-02 16:52:31] Fine-tune started
[2023-06-02 17:09:32] Completed epoch 1/4
[2023-06-02 17:36:31] Completed epoch 3/4
[2023-06-02 17:50:29] Uploaded model: davinci:ft-personal-2023-06-02-09-50-29
[2023-06-02 17:50:31] Uploaded result file: file-WoUhP6XjpKGyZznvthkviMIf
[2023-06-02 17:50:31] Fine-tune succeeded
```

# Case Study – OpenAI

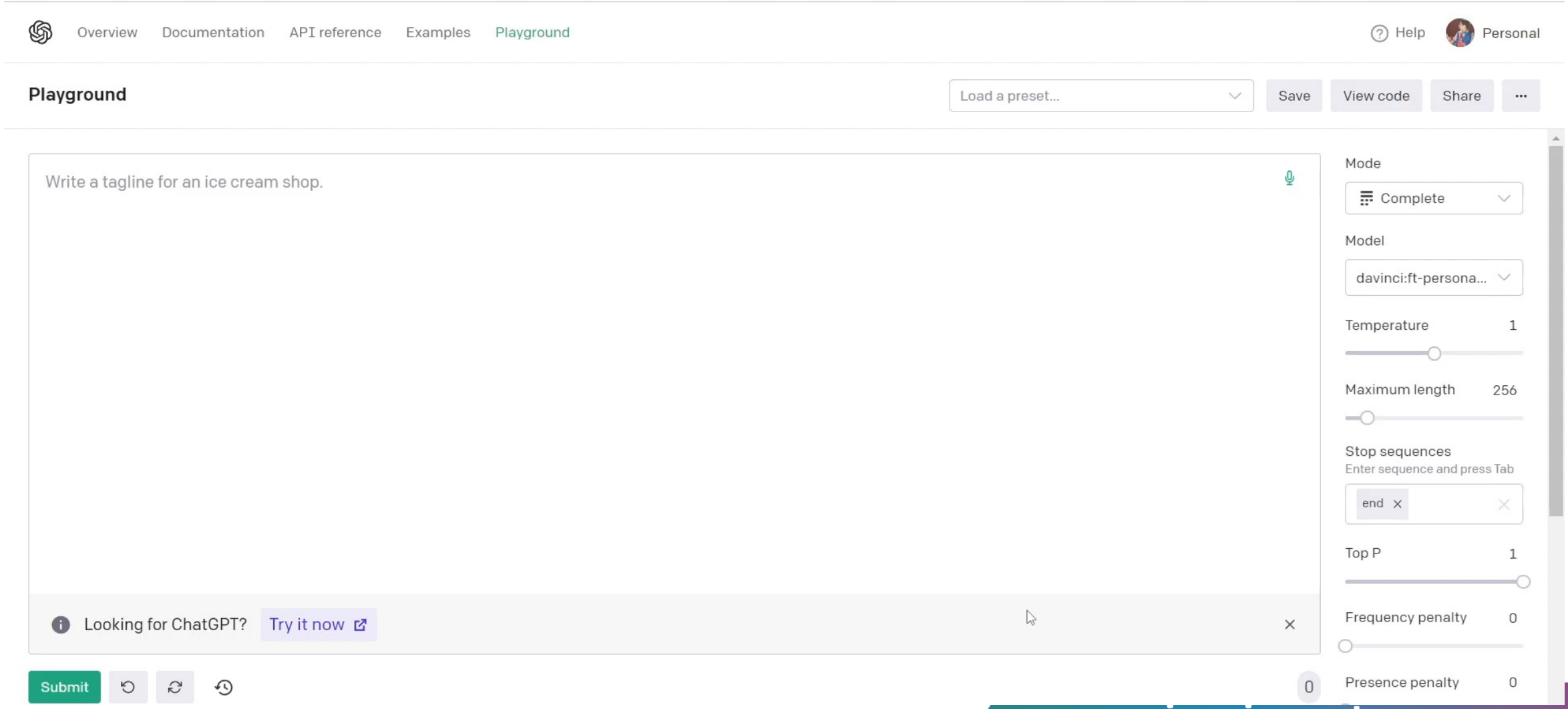
**Step 7: trace the progress (Detail Classification): 1348 pairs**

```
openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>
```

```
[2023-06-02 19:33:33] Created fine-tune: ft-t3V3yklK1lTZxy0CWFbiP2nG
[2023-06-02 21:05:00] Fine-tune costs $25.71
[2023-06-02 21:05:00] Fine-tune enqueued. Queue number: 2
[2023-06-02 21:05:03] Fine-tune is in the queue. Queue number: 1
[2023-06-02 21:05:05] Fine-tune is in the queue. Queue number: 0
[2023-06-02 21:05:09] Fine-tune started
[2023-06-02 21:19:43] Completed epoch 1/4
[2023-06-02 21:43:14] Completed epoch 3/4
[2023-06-02 21:55:55] Uploaded model: davinci:ft-personal-2023-06-02-13-55-55
[2023-06-02 21:55:57] Uploaded result file: file-8IorcwLDluWd7JbRwMQGOm0V
[2023-06-02 21:55:57] Fine-tune succeeded
```

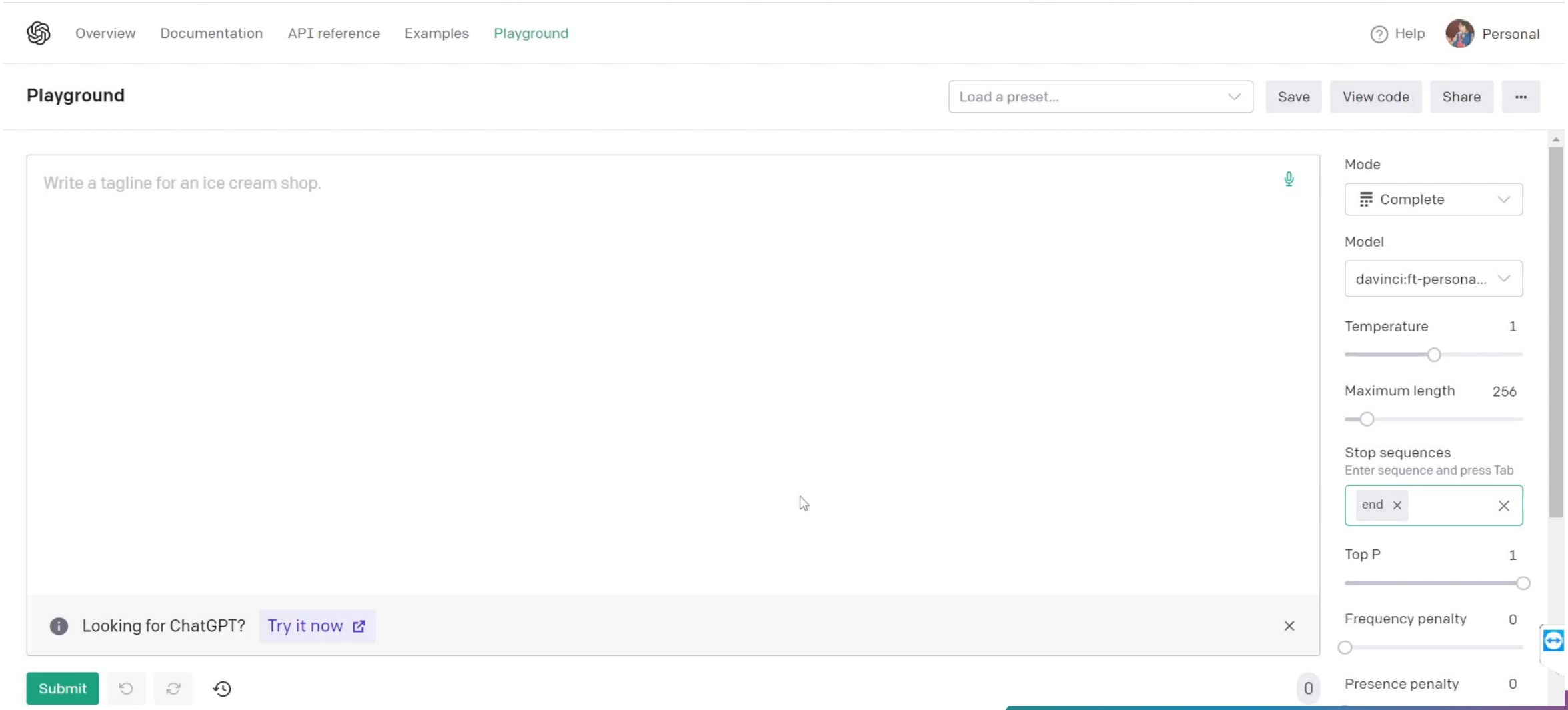
# Case Study – OpenAI

## Scenario 1: Garbage message or NOT



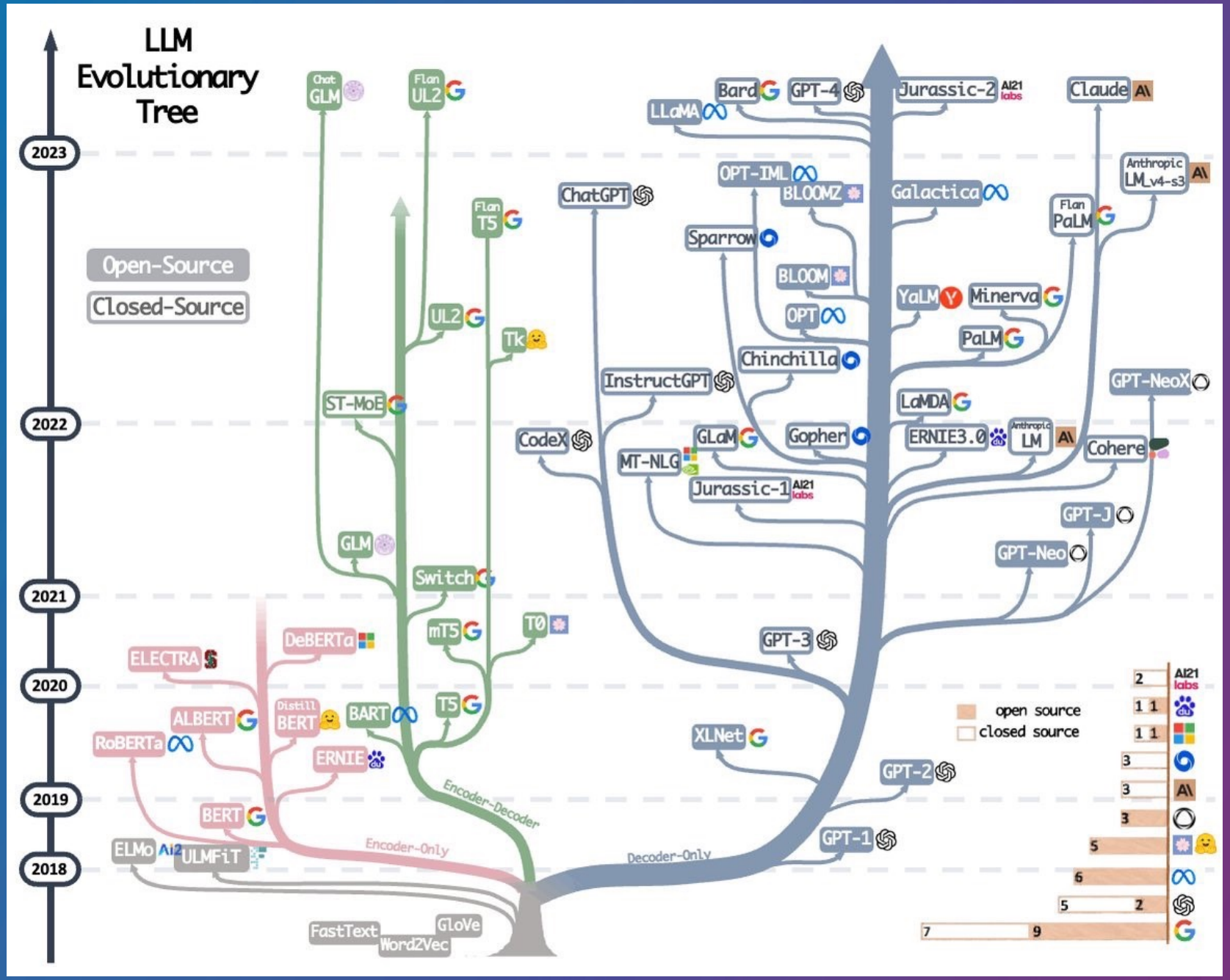
# Case Study – OpenAI

## Scenario 2: Detail Classification



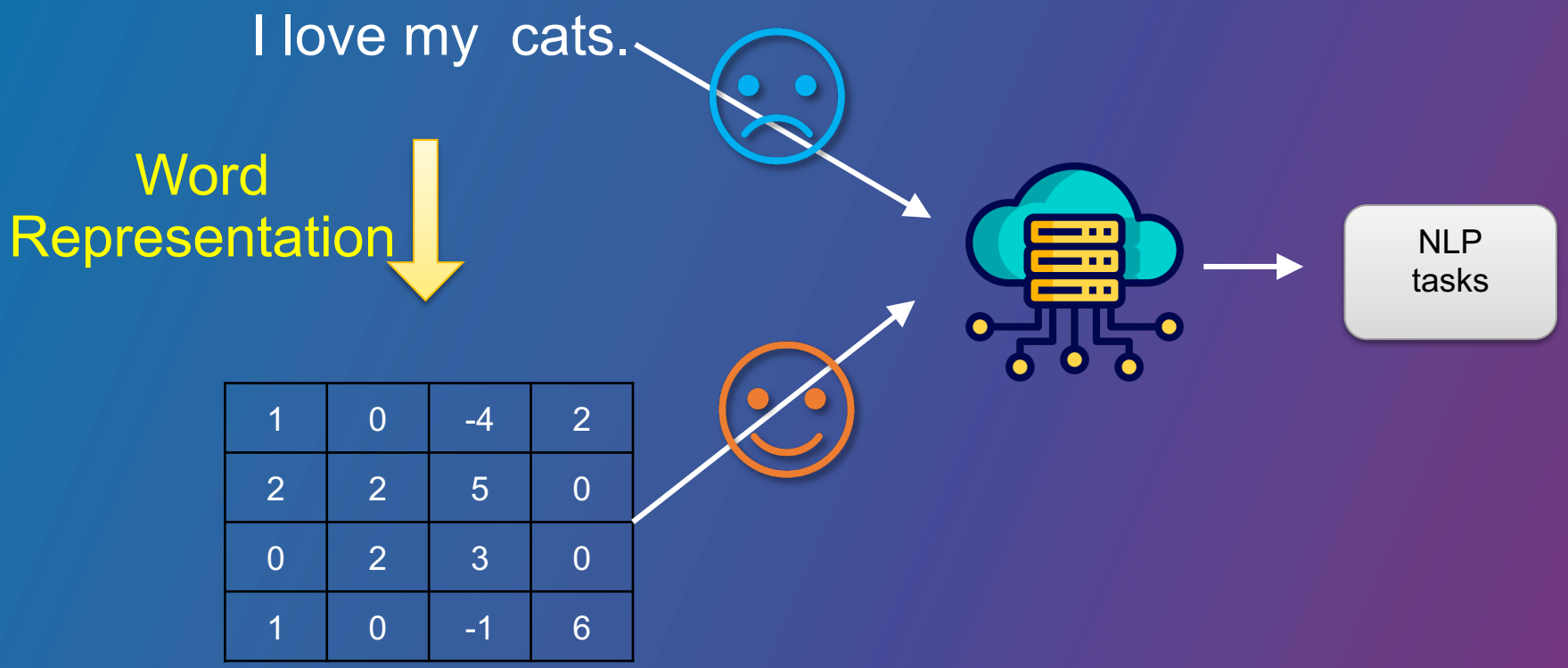
# LoRa for EMIC

- **Model:**
  - **minlik/chinese-llama-plus-7b-merged**
- **Datasets:**
  - **EMIC**
- **Hyperparameters in LoRa:**
  - **r=16, #attention heads**
  - **lora_alpha=32, # scaling factor for the weight matrices**
  - **lora_dropout=0.05,**
  - **bias="none",**
- **Hyperparameters in training:**
  - **per_device_train_batch_size=1,**
  - **gradient_accumulation_steps=4,**
  - **warmup_steps=100,**
  - **max_steps=3000,**
  - **learning_rate=2e-4,**
  - **optim='adamw_torch',**



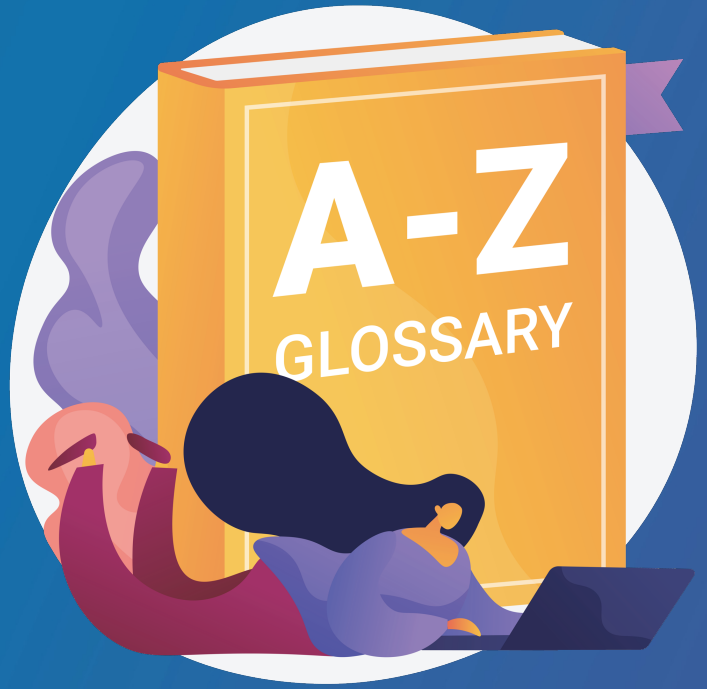

https://github.com/Mooler0410/LLMsPracticalGuide

# The ability of computer to understand human language




https://medium.com/nlplanet/what-tasks-can-i-solve-with-nlp-today-1b1823cc8cdf

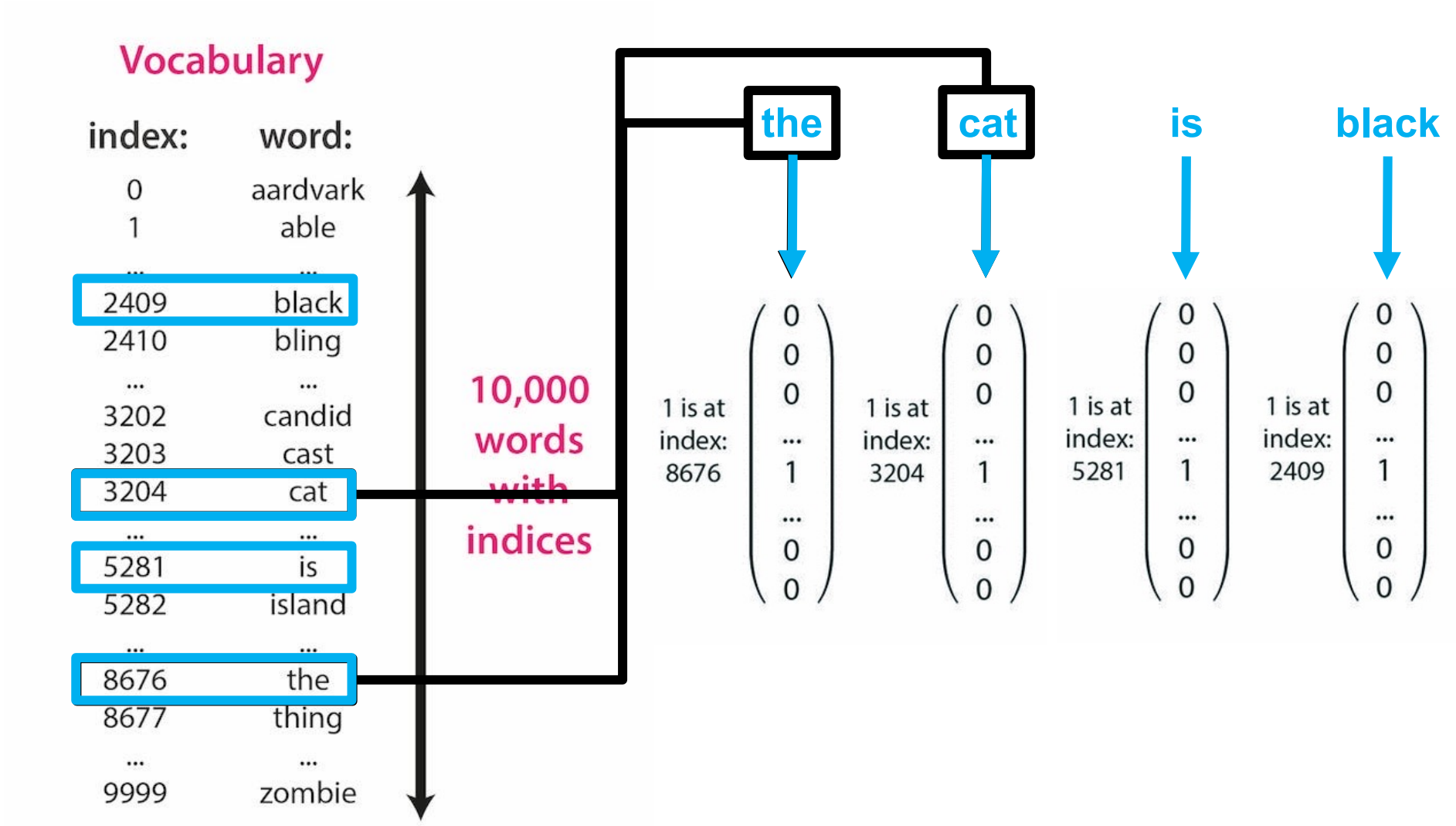
# Word Representation




## Corpus based

- One-hot Representation
- Distributional Representation

# One-Hot Encoding

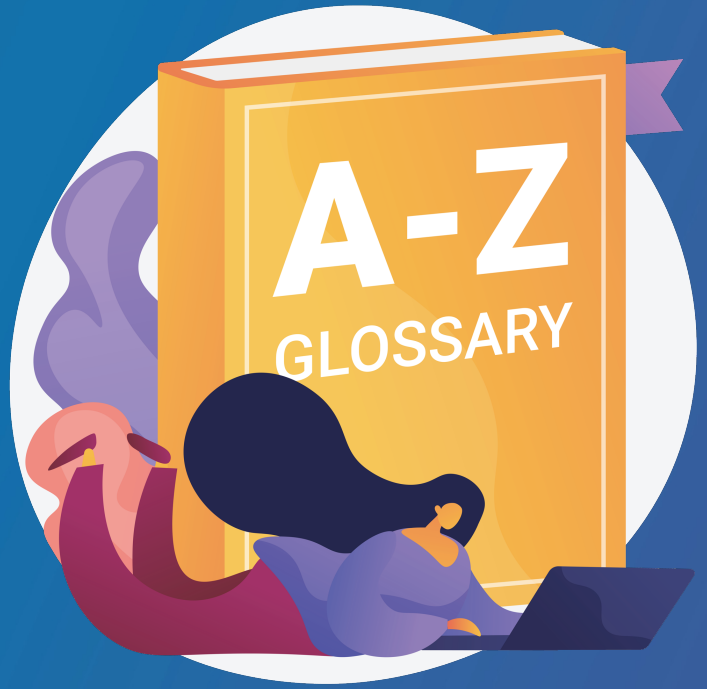



https://towardsdatascience.com/why-do-we-use-embeddings-in-nlp-2f20e1b632d2

# However , the limitations are…

- High-dimensional representation – vector sparsity
- Inability to capture semantic similarity
- Loss of word order information

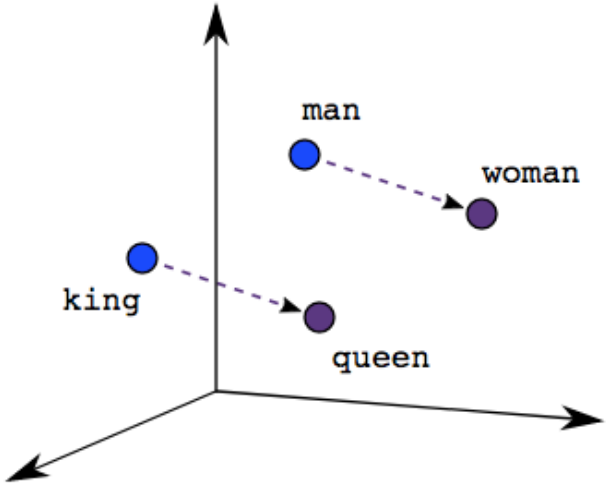
## Word Representation



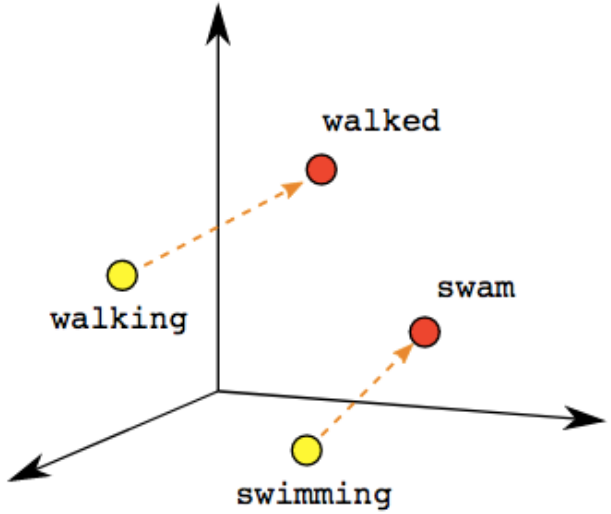

## Corpus based

- One-hot Representation
- Distributional Representation

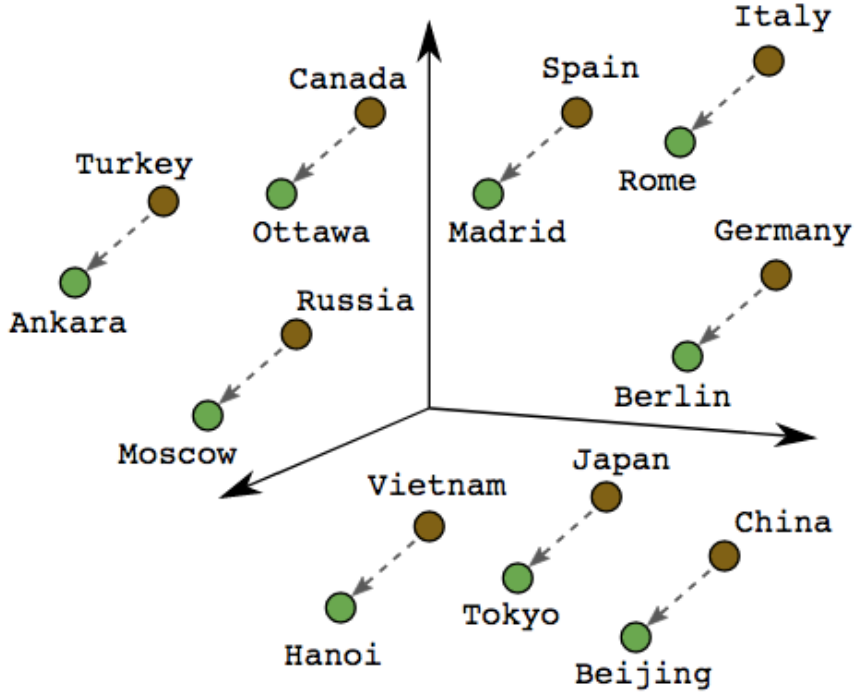
You shall know a word by the company it keeps (Firth, 1957)




Male-Female




Verb Tense




Country-Capital



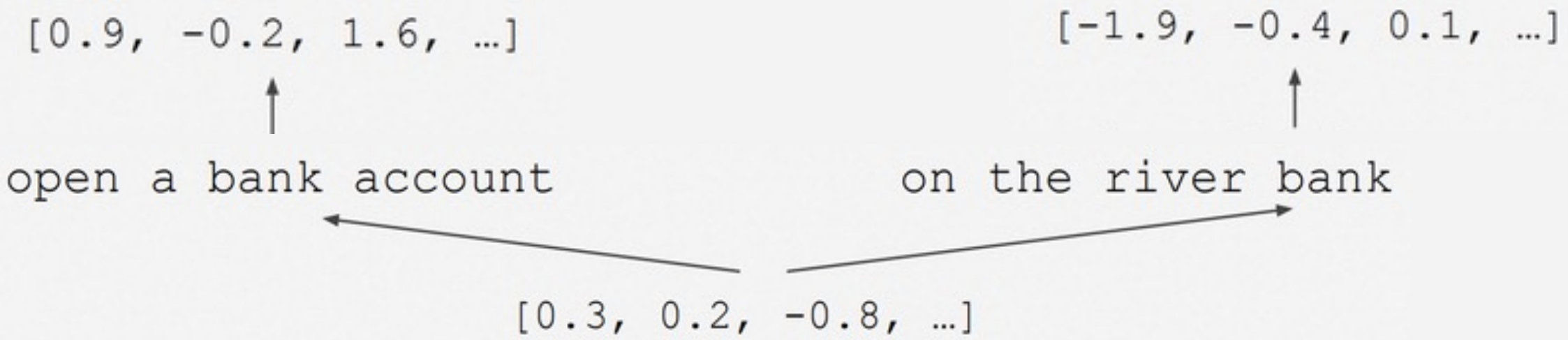
https://towardsdatascience.com/a-guide-to-word-embeddings-8a23817ab60f

# Word2Vec (2013)

sliding window

The dog can’t stop chasing the cat on the street.

## contextual dependent

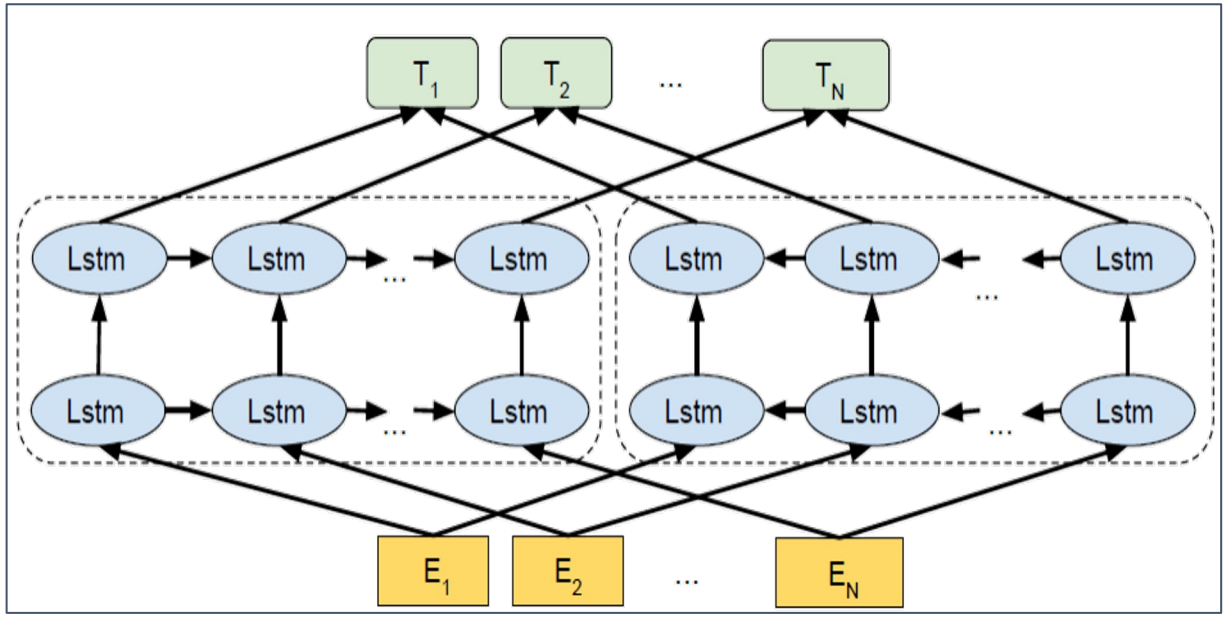



## contextual independent

## Deep contextualized word representations (ELMo)

**RNN based**

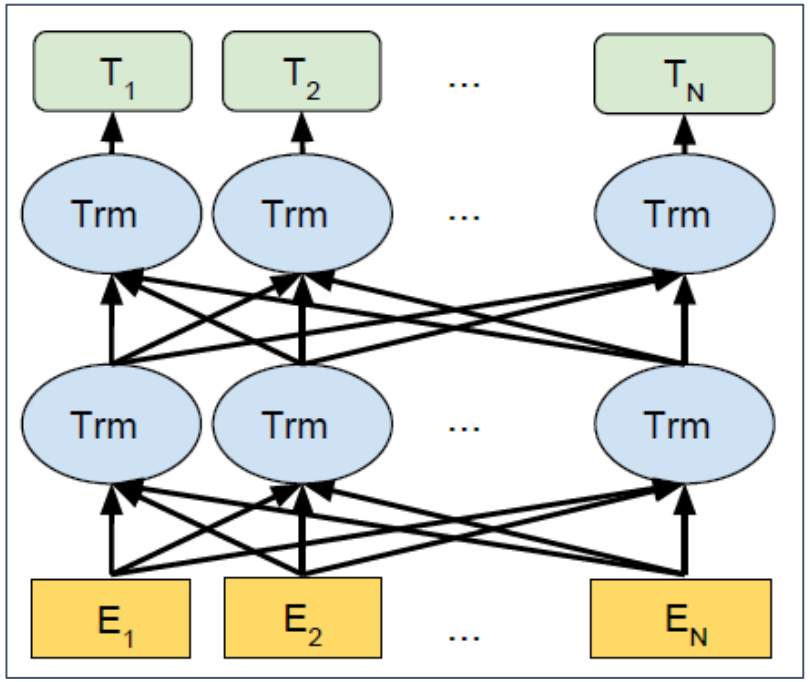
- **Bi-LSTM : forward + backward**
- **predicting next word**




## Bidirectional Encoder Representations from Transformers (BERT)

**Transformer based**

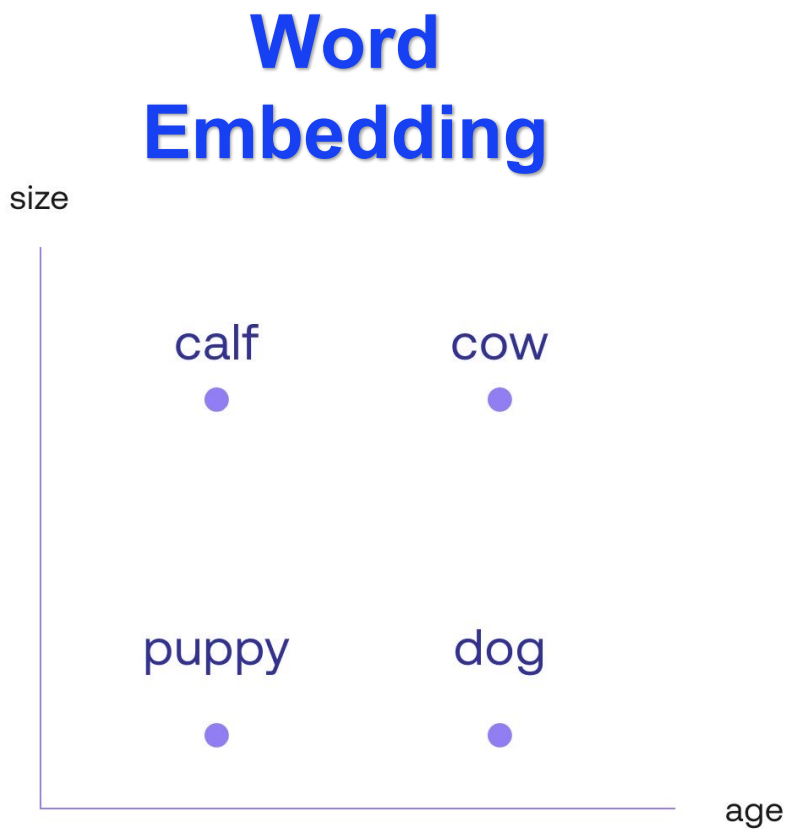
- **self-attention**
- **predicting masked words and next sentence relationships**




https://arxiv.org/pdf/1810.04805.pdf

https://arxiv.org/pdf/1802.05365.pdf

# Variation of Embeddings

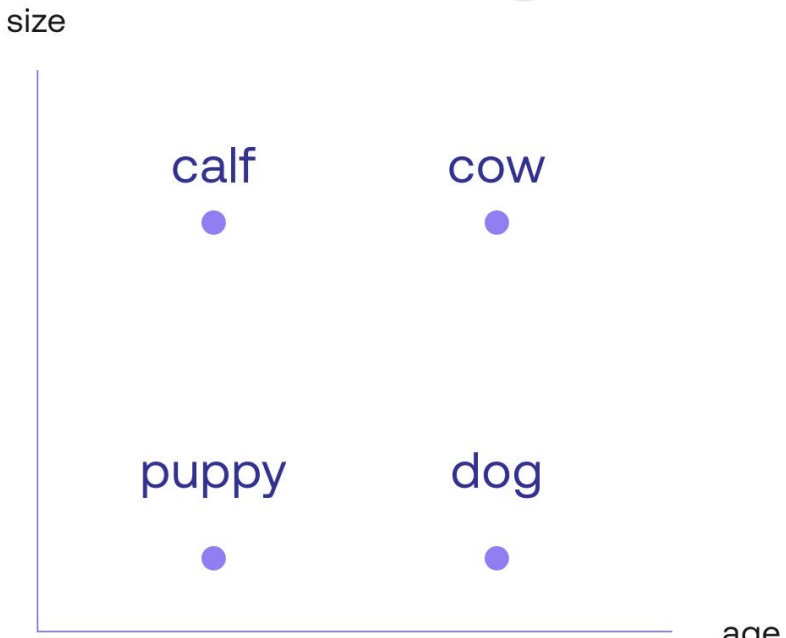




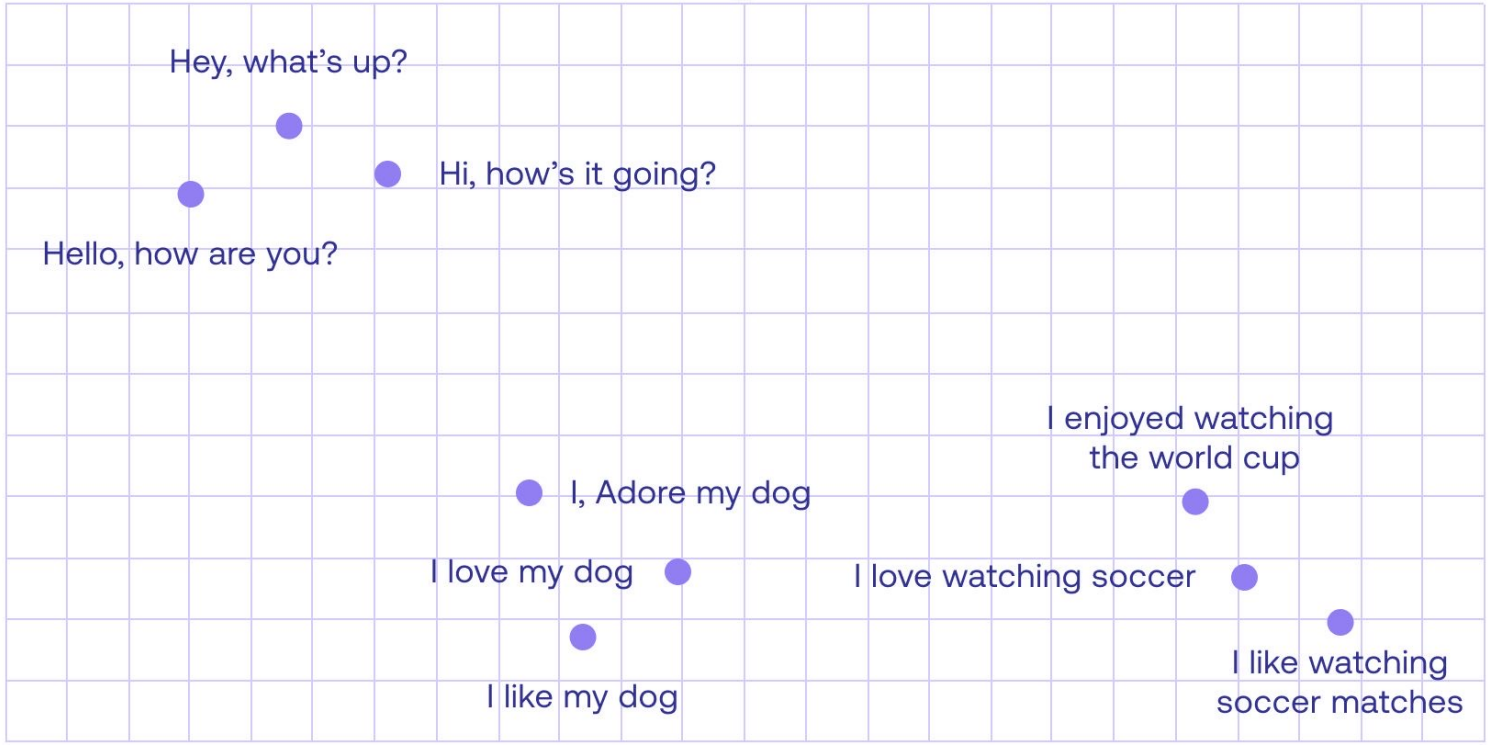

https://txt.cohere.com/sentence-word-embeddings/

# Variation of Embeddings

## Word Embedding

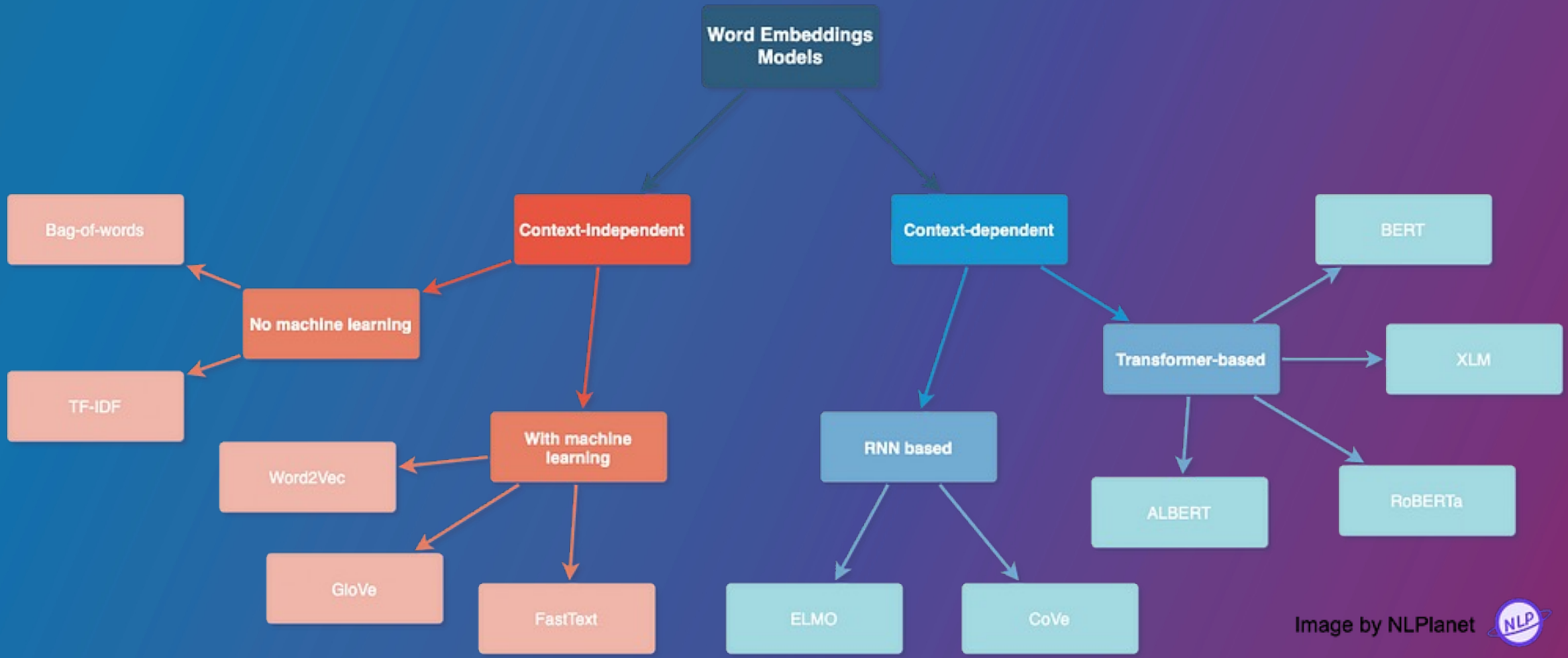



## Sentence Embedding




https://txt.cohere.com/sentence-word-embeddings/

# Much left unsaid …



# How to use word-embeddings

- Search
- Clustering
- Recommendations
- Anomaly detection
- Diversity measurement
- Classification

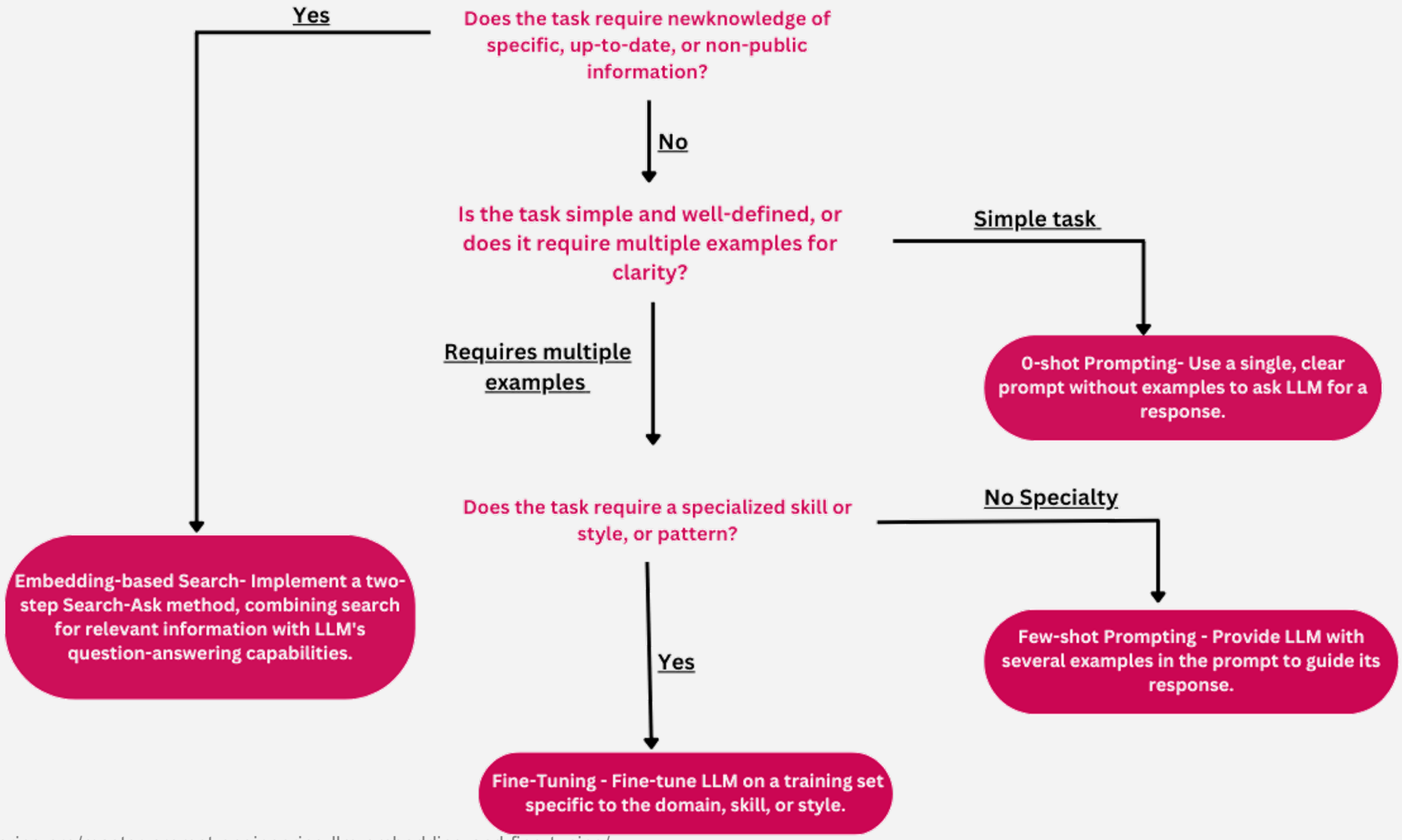


https://platform.openai.com/docs/guides/embeddings/use-cases

Prompt Engineering
PromptEngineering.org

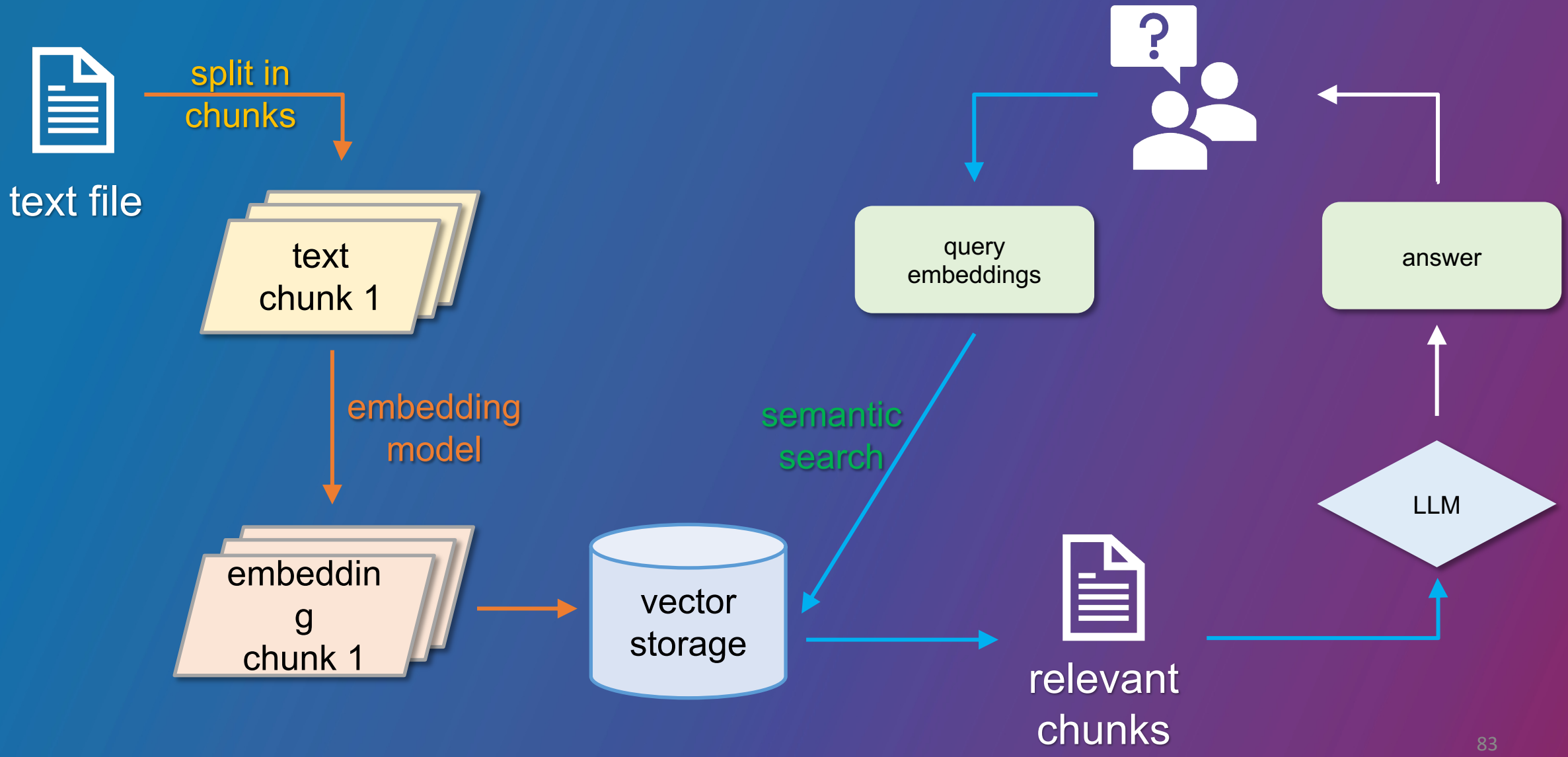
# Choosing the Right LLM Strategy

## 0-shot vs Few-shot vs Fine-tuning vs Embedding

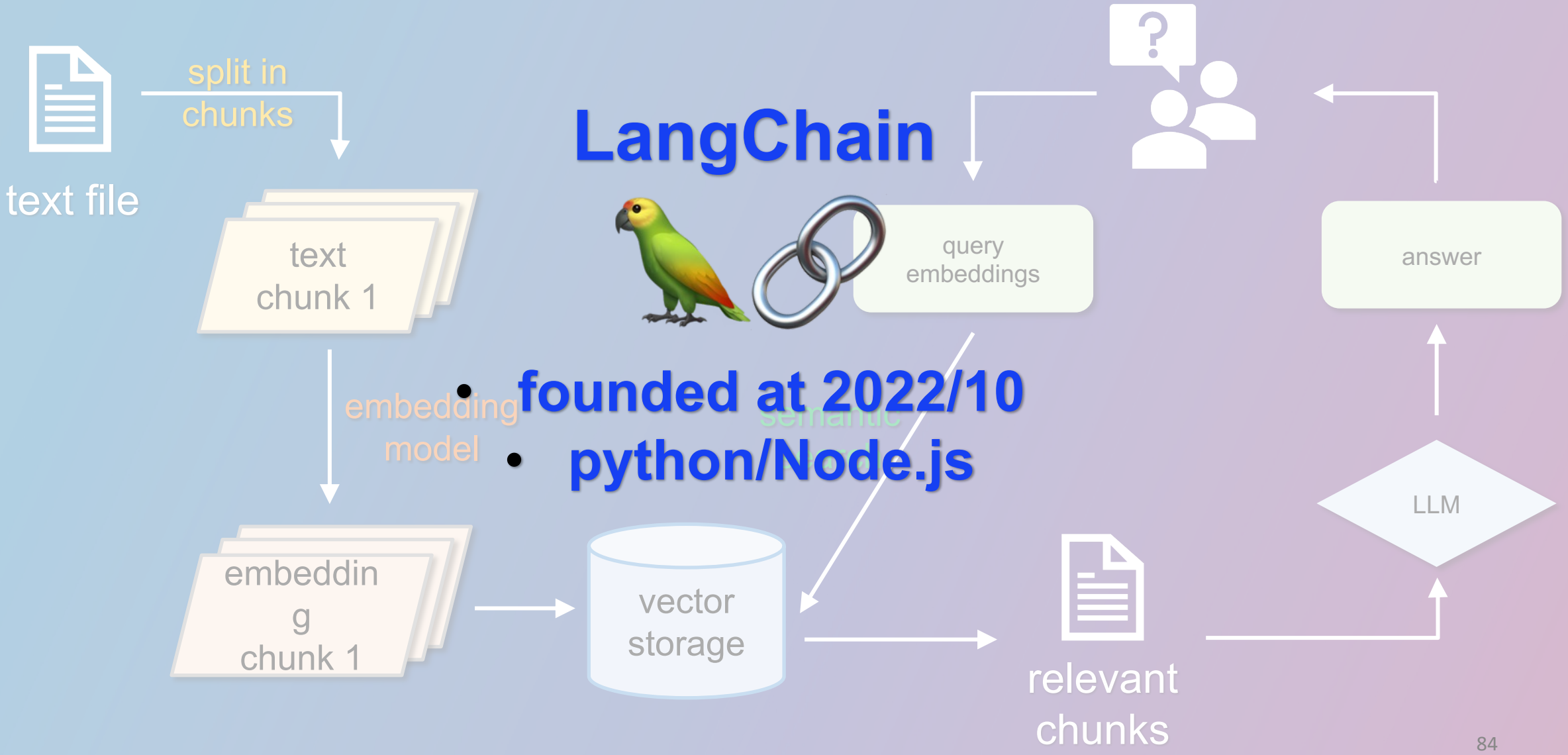



https://www.promptengineering.org/master-prompt-engineering-llm-embedding-and-fine-tuning/

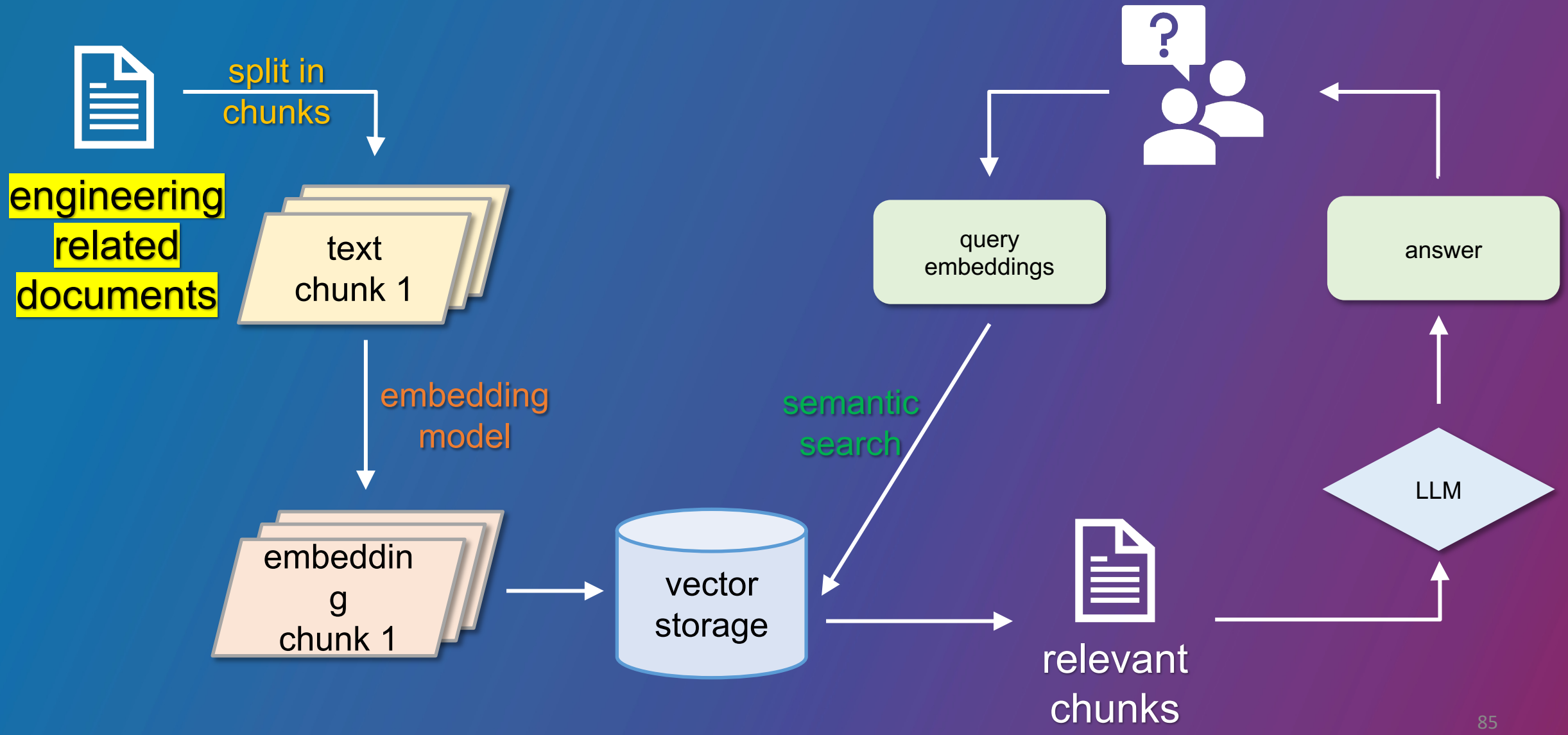
# Embedding-based Application - QA



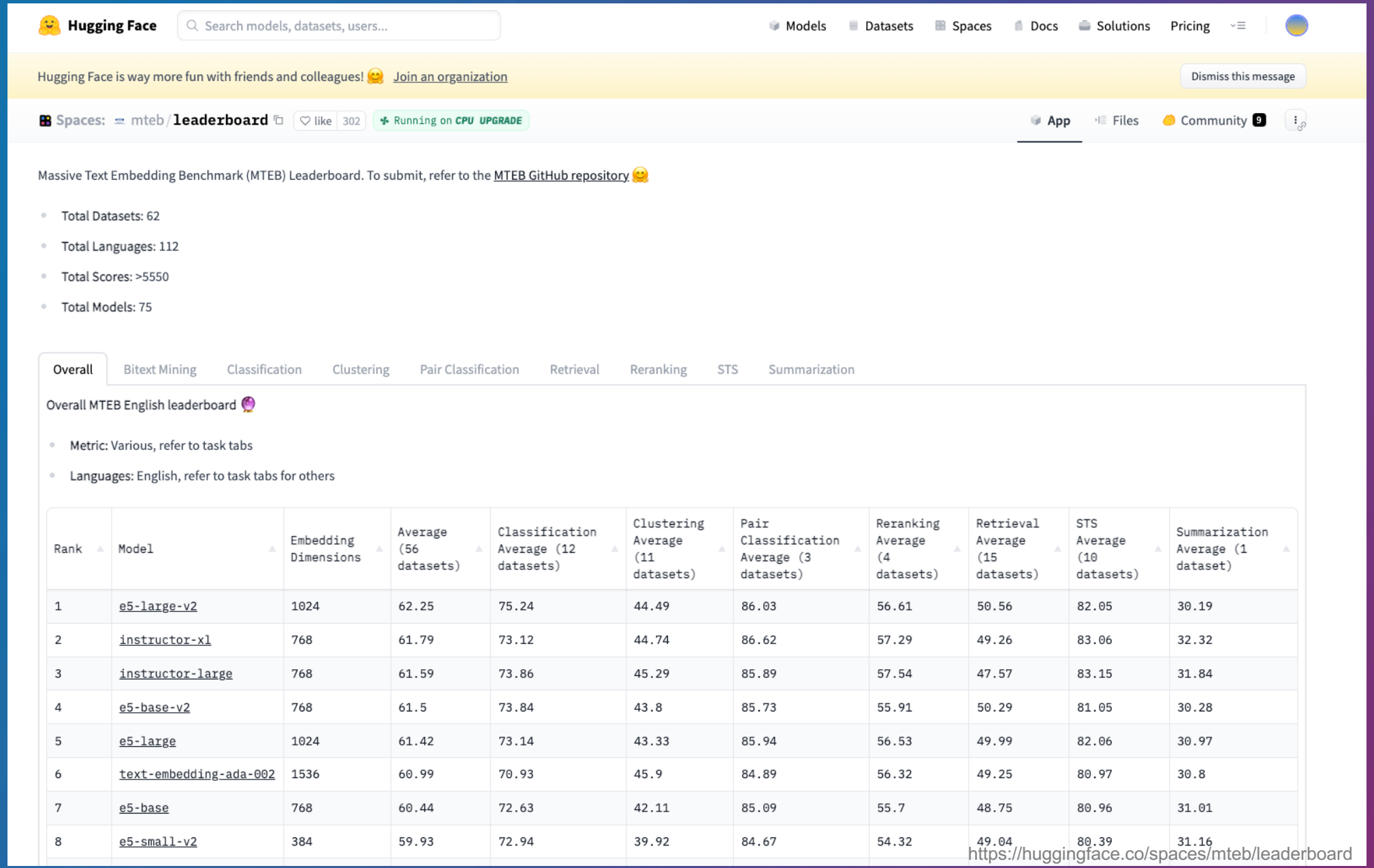
# Embedding-based Application



# Embedding-based Application – QA with documents



# model selection




| Rank | Model | Embedding Dimensions | Average (56 datasets) | Classification Average (12 datasets) | Clustering Average (11 datasets) | Pair Classification Average (3 datasets) | Reranking Average (4 datasets) | Retrieval Average (15 datasets) | STS Average (10 datasets) | Summarization Average (1 dataset) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | e5-large-v2 | 1024 | 62.25 | 75.24 | 44.49 | 86.03 | 56.61 | 50.56 | 82.05 | 30.19 |
| 2 | instructor-xl | 768 | 61.79 | 73.12 | 44.74 | 86.62 | 57.29 | 49.26 | 83.06 | 32.32 |
| 3 | instructor-large | 768 | 61.59 | 73.86 | 45.29 | 85.89 | 57.54 | 47.57 | 83.15 | 31.84 |
| 4 | e5-base-v2 | 768 | 61.5 | 73.84 | 43.8 | 85.73 | 55.91 | 50.29 | 81.05 | 30.28 |
| 5 | e5-large | 1024 | 61.42 | 73.14 | 43.33 | 85.94 | 56.53 | 49.99 | 82.06 | 30.97 |
| 6 | text-embedding-ada-002 | 1536 | 60.99 | 70.93 | 45.9 | 84.89 | 56.32 | 49.25 | 80.97 | 30.8 |
| 7 | e5-base | 768 | 60.44 | 72.63 | 42.11 | 85.09 | 55.7 | 48.75 | 80.96 | 31.01 |
| 8 | e5-small-v2 | 384 | 59.93 | 72.94 | 39.92 | 84.67 | 54.32 | 49.04 | 80.39 | 31.16 |

https://huggingface.co/spaces/mteb/leaderboard

```
from langchain.document_loaders import *
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import Chroma, Pinecone
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.embeddings.huggingface import HuggingFaceEmbeddings,HuggingFaceInstructEmbeddings,
from langchain.embeddings import HuggingFaceEmbeddings, SentenceTransformerEmbeddings
from langchain.chains.question_answering import load_qa_chain
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate

loader = UnstructuredWordDocumentLoader(('./中興施工規範範例檔案/00001-道碴軌道施工規範-1.0.doc'))
data = loader.load()

text_splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=200)
texts = text_splitter.split_documents(data)
print (f'Now you have {len(texts)} documents')

embeddings=HuggingFaceEmbeddings(model_name='shibing624/text2vec-base-chinese’)


db = Chroma.from_documents(texts, embeddings)
retriever = db.as_retriever(search_type="similarity", search_kwargs={"k":2})


query = input(“Question:”)
docs = retriever.get_relevant_documents(query)
```

```
from langchain.document_loaders import *
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import Chroma, Pinecone
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.embeddings.huggingface import HuggingFaceEmbeddings,HuggingFaceInstructEmbeddings,
from langchain.embeddings import HuggingFaceEmbeddings, SentenceTransformerEmbeddings
from langchain.chains.question_answering import load_qa_chain
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate

loader = UnstructuredWordDocumentLoader(('./中興施工規範範例檔案/00001-道碴軌道施工規範-1.0.doc'))
data = loader.load()

text_splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=200)
texts = text_splitter.split_documents(data)
print (f'Now you have {len(texts)} documents')


embeddings=HuggingFaceEmbeddings(model_name='shibing624/text2vec-base-chinese’)


db = Chroma.from_documents(texts, embeddings)
retriever = db.as_retriever(search_type="similarity", search_kwargs={"k":2})


query = input(“Question:”)
docs = retriever.get_relevant_documents(query)
```

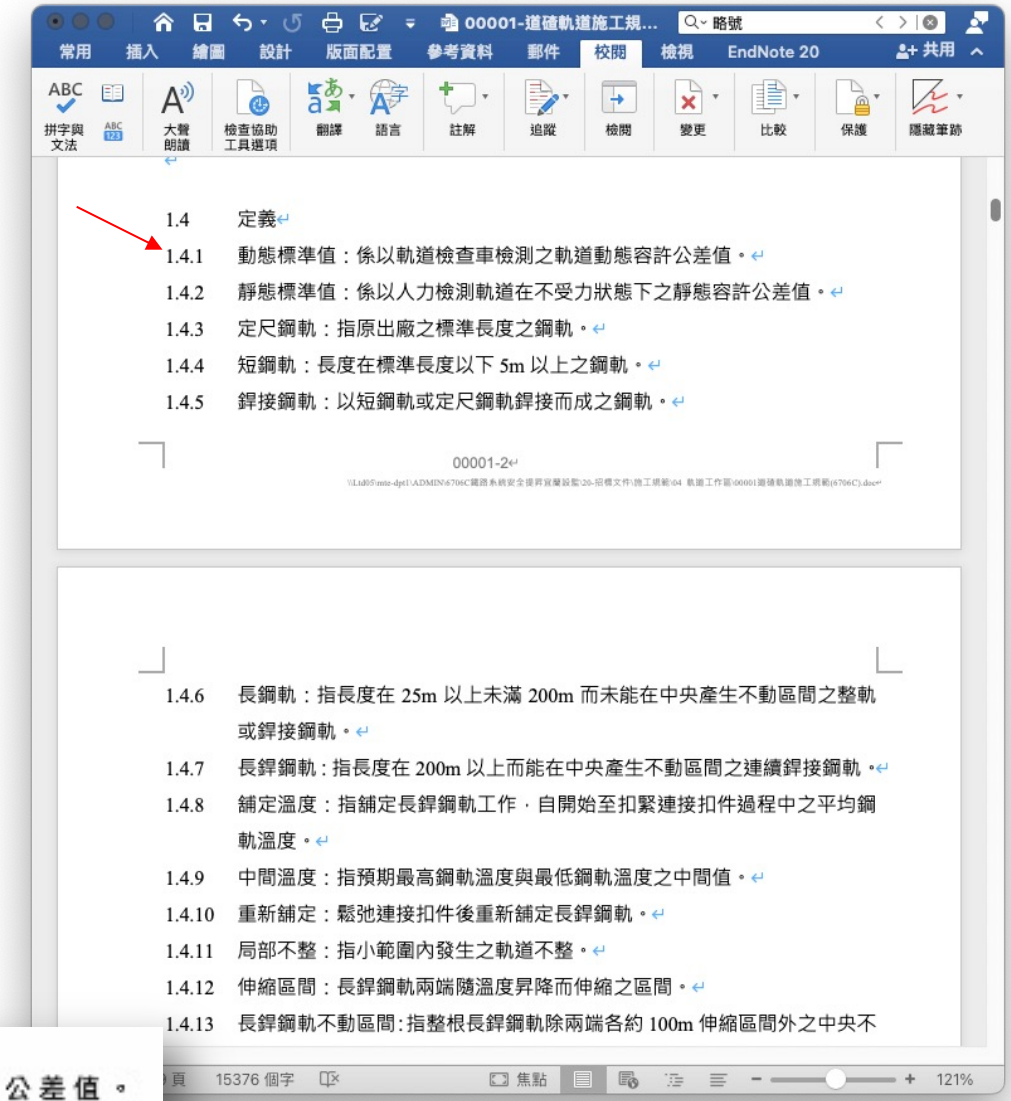
```
db = Chroma.from_documents(texts, embeddings)
retriever = db.as_retriever(search_type="similarity", search_kwargs={"k":2})


query = input(“Question:”)
docs = retriever.get_relevant_documents(query)
```

```
prompt_template = """請注意：請指根據本段輸入文字訊息的內容進行回答，如果query與提供的內容無關，請回答
“我不知道”，另外也不要回答無關答案：
Context: {context}
Question: {question}
Answer:
請以中文做問答"""
# """
PROMPT = PromptTemplate(template=prompt_template, input_variables=["context", "question"])

qa = load_qa_chain(ChatOpenAI(openai_api_key=OPENAI_API_KEY,temperature=0),
                           chain_type="stuff",
                           prompt=PROMPT,
                           verbose=True)
result = qa({"input_documents": docs, "question": query}, return_only_outputs=False)
print(result["output_text"])
```


embedding — python ◂ python BookQA_engineer.py — 114×45
(pytorch-m1) hamigua:embedding cengqianyu$ python BookQA_engineer.py
Could not import azure.core python package.
convert /Users/cengqianyu/Google Drive - a/NTU/PHD/2022spring_AiSeminar/embedding/中興施工規範範例檔案/00001-道碴軌道施工規範-1.0.doc -> /private/var/folders/by/vzsz3lx903j6drt3wzdxhj4m0000gn/T/tmp3wipqbid/00001-道碴軌道施工規範-1.0.docx using filter : MS Word 2007 XML
You have 1 document(s) in your data
Now you have 352 documents
Question:動態標準值在哪些章節有提到？他的定義是什麼
> Finished chain.
Answer:動態標準值在1.4.1定義章節有提到，其定義為以軌道檢查車檢測之軌道動態容許公差值。


1.4 定義
1.4.1 動態標準值：係以軌道檢查車檢測之軌道動態容許公差值。
1.4.2 靜態標準值：係以人力檢測軌道在不受力狀態下之靜態容許公差值。
1.4.3 定尺鋼軌：指原出廠之標準長度之鋼軌。
1.4.4 短鋼軌：長度在標準長度以下 5m 以上之鋼軌。
1.4.5 銲接鋼軌：以短鋼軌或定尺鋼軌銲接而成之鋼軌。
00001-2
1.4.6 長鋼軌：指長度在 25m 以上未滿 200m 而未能在中央產生不動區間之整軌或銲接鋼軌。
1.4.7 長銲鋼軌：指長度在 200m 以上而能在中央產生不動區間之連續銲接鋼軌。
1.4.8 鋪定溫度：指鋪定長銲鋼軌工作，自開始至扣緊連接扣件過程中之平均鋼軌溫度。
1.4.9 中間溫度：指預期最高鋼軌溫度與最低鋼軌溫度之中間值。
1.4.10 重新鋪定：鬆弛連接扣件後重新鋪定長銲鋼軌。
1.4.11 局部不整：指小範圍內發生之軌道不整。
1.4.12 伸縮區間：長銲鋼軌兩端隨溫度昇降而伸縮之區間。
1.4.13 長銲鋼軌不動區間：指整根長銲鋼軌除兩端各約 100m 伸縮區間外之中央不
15376 個字
121%

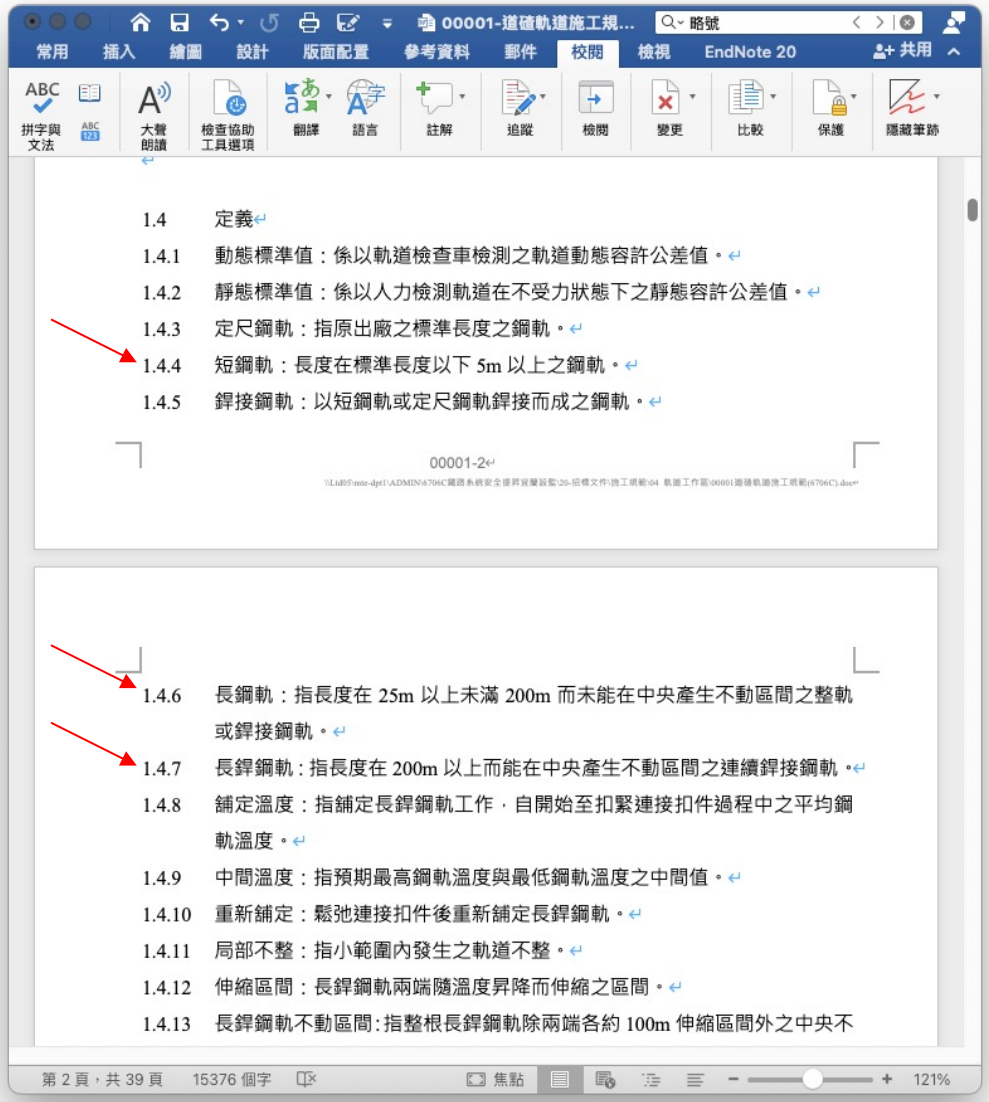
**Question: 請分別告訴我短鋼軌、長鋼軌以及長銲鋼軌的定義是什麼？出現在哪幾章節**

*text2vec-base-chinese*

Answer:**短鋼軌是指長度在標準長度以下5m以上之鋼軌，出現在1.4.4章節。**長鋼軌是指長度在25m以上未滿200m而未能在中央產生不動區間之整軌或銲接鋼軌，出現在1.4.6章節。長銲鋼軌是指長度在200m以上而能在中央產生不動區間之連續銲接鋼軌，出現在1.4.7章節。

*OpenAI- text-embedding-ada-002*

Answer:**短鋼軌是指定尺鋼軌或短斷鋼軌，銲接鋼軌是由短鋼軌或定尺鋼軌銲接而成，這些定義出現在第1章的1.4.5節。**長鋼軌是指長度在25m以上未滿200m而未能在中央產生不動區間之整軌或銲接鋼軌，長銲鋼軌是指長度在200m以上而能在中央產生不動區間之連續銲接鋼軌，這些定義出現在第1章的1.4.6節和1.4.7節。



# QA with youtube video

- **Six behaviors to increase your confidence | Emily Jaenson | TEDxReno**

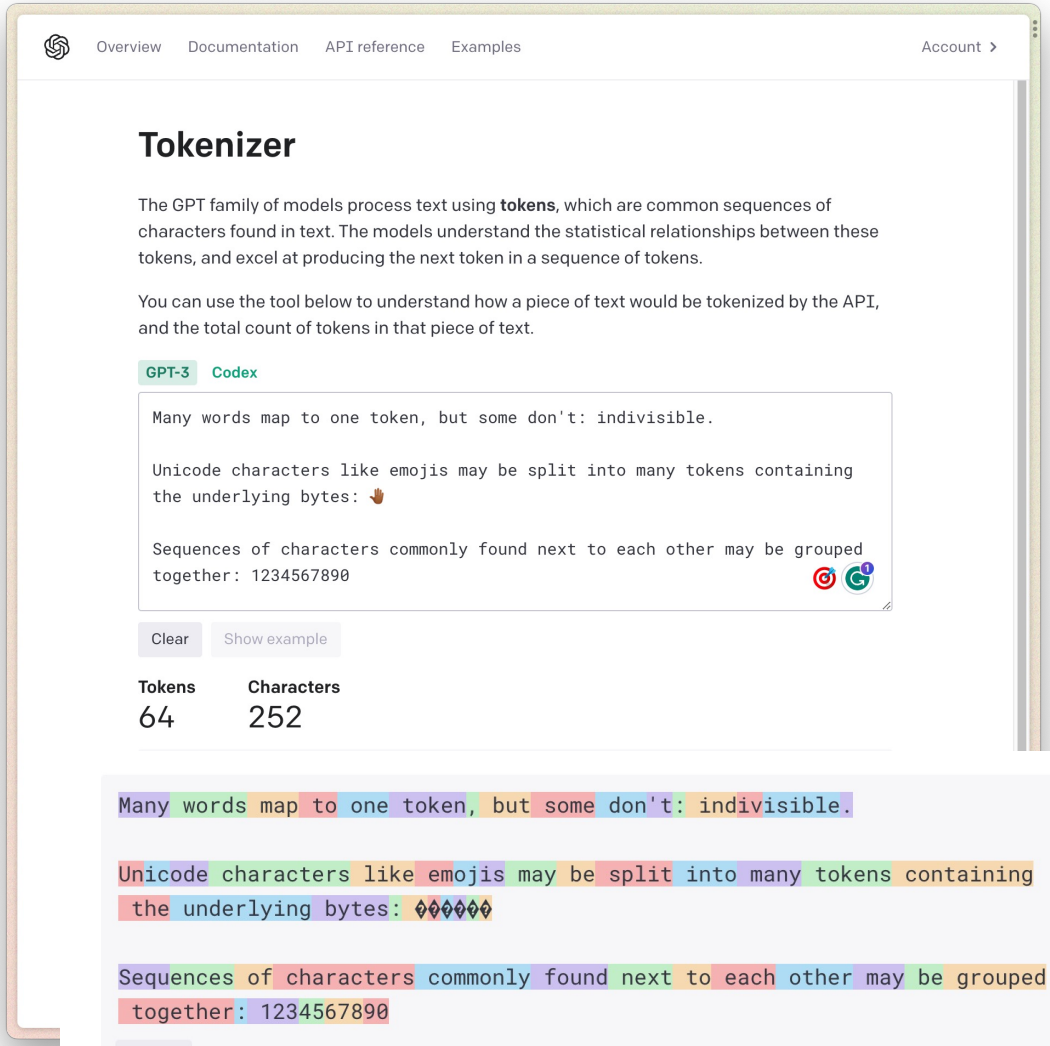
```
loader = YoutubeLoader.from_youtube_url('https://www.youtube.com/watch?v=IitIl2C3Iy8')
```

```
ArxivLoader, DiscordChatLoader, EverNoteLoader, GoogleDriveLoader, NotionDBLoader…
```

```
Question: what are the six behaviors?

Answer:
1. Count Yourself In
2. Give Yourself 20 Seconds of Courage
3. Take a Seat at the Table
4. Cheer for Other People's Success
5. Bolster Your Confidence for a New Activity Through Your Already Great Performance in Another
6. Celebrate Constantly
```

# TIKTOKEN – token calculator

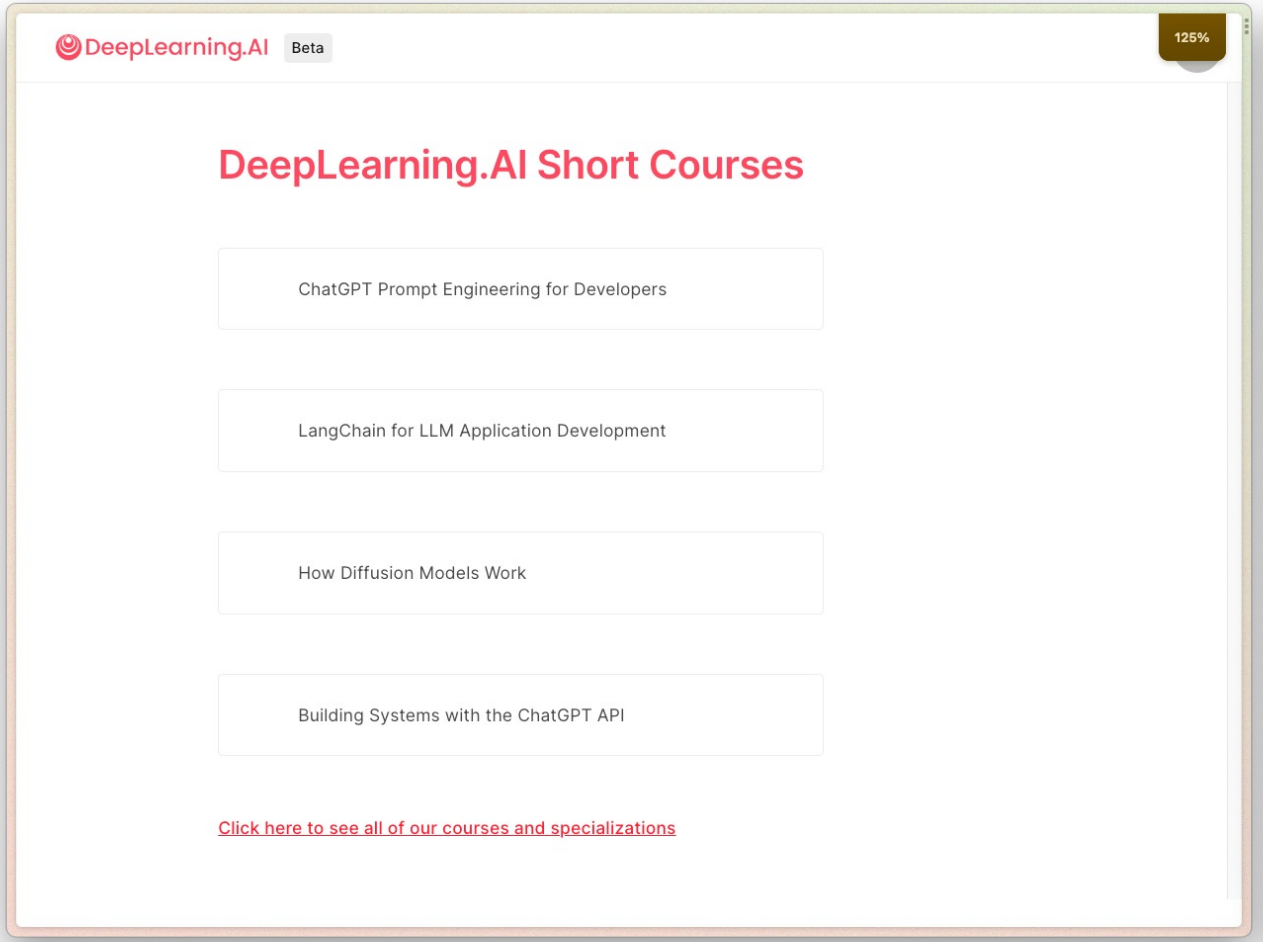



```
MODEL_TO_ENCODING: dict[str, str] = {
# chat
"gpt-4": "cl100k_base",
"gpt-3.5-turbo": "cl100k_base",
# text
"text-davinci-003": "p50k_base",
"text-davinci-002": "p50k_base",
"text-davinci-001": "r50k_base",
"text-curie-001": "r50k_base",
"text-babbage-001": "r50k_base",
"text-ada-001": "r50k_base",
"davinci": "r50k_base",
"curie": "r50k_base",
"babbage": "r50k_base",
"ada": "r50k_base",
}
```

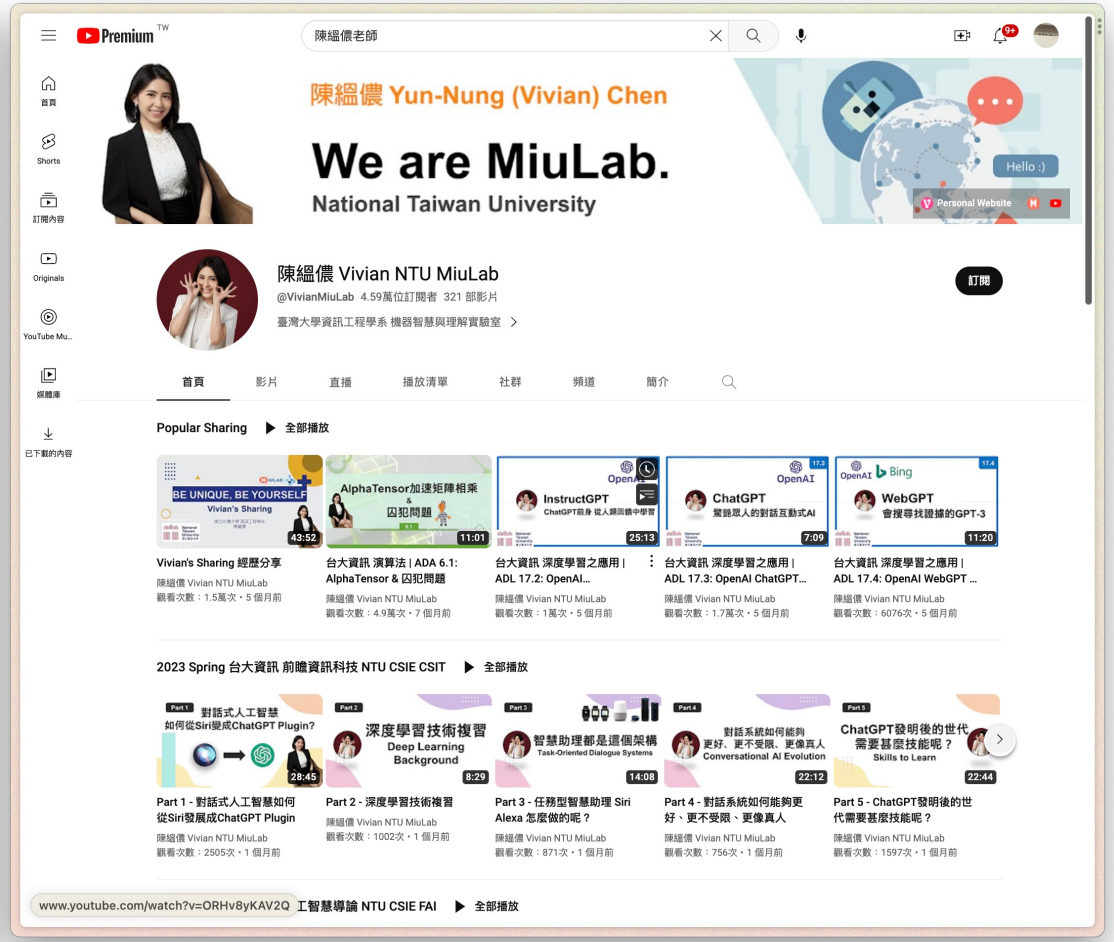
```
for i in texts:
t=i.page_content

encoding = tiktoken.get_encoding("cl100k_base")
token_len = len(encoding.encode(t))
```

# Online Courses




https://learn.deeplearning.ai/




https://www.youtube.com/@VivianMiuLab


**Hope you enjoy the sharing!**



https://towardsai.net/p/machine-learning/you-can-no-longer-fail-to-understand-how-to-use-large-language-models