# NLP簡介和Tokenizer

黃志勝（Tommy Huang）

義隆電子 人工智慧研發部

國立陽明交通大學 AI學院 合聘助理教授

國立台北科技大學 電資學院合聘助理教授

# NLP（自然語言處理）

自然語言處理 (NLP) 是電腦程式用來解釋人類語言的一種方法。

**情緒分析**：NLP 可以幫助解釋大量使用者評論、社交媒體貼文或客戶服務請求

**虛擬助理**：NLP 對於理解 Siri、Alexa 或 Cortana 等助理使用者的請求至關重要

**搜尋引擎**：NLP 幫助搜尋引擎更好地理解簡單的單字查詢和句子或問題形式查詢背後的搜尋意圖，以及解釋查詢中的拼字錯誤或其他人為錯誤

**翻譯**：NLP 可以幫助理解內容並將其從一種語言翻譯成另一種語言

**內容審核**：NLP 可以透過解釋使用者產生文字的含義來幫助標記潛在有害或令人反感的內容

# 大型語言模型 (LLM)

ChatGPT 4o ⌄                                    ⬆ 分享

> 幫我出一份機器學習的基本考題，總共五題，一題20分，總共100分

這是一份關於機器學習的基本考題，適合評估考生對機器學習核心概念的理解。每題20分，總分100分。

---

**機器學習基本考題**

**1. 監督學習與非監督學習的區別（20分）**
請簡要解釋什麼是監督學習和非監督學習，並提供每種學習方法的一個實際應用例子。

**2. 過擬合（Overfitting）與欠擬合（Underfitting）的概念（20分）**
什麼是過擬合和欠擬合？如何避免模型過擬合？請列出至少兩種方法來解決過擬合的問題。
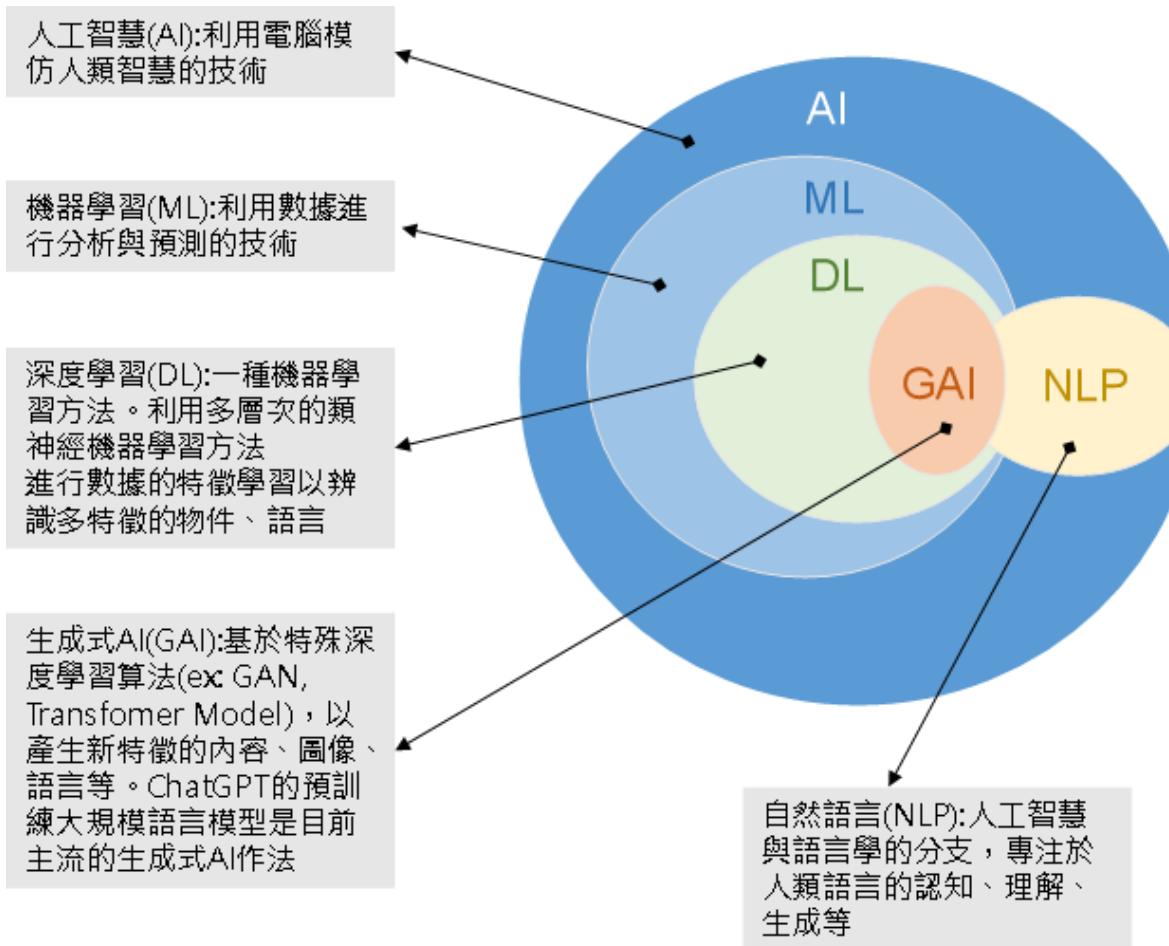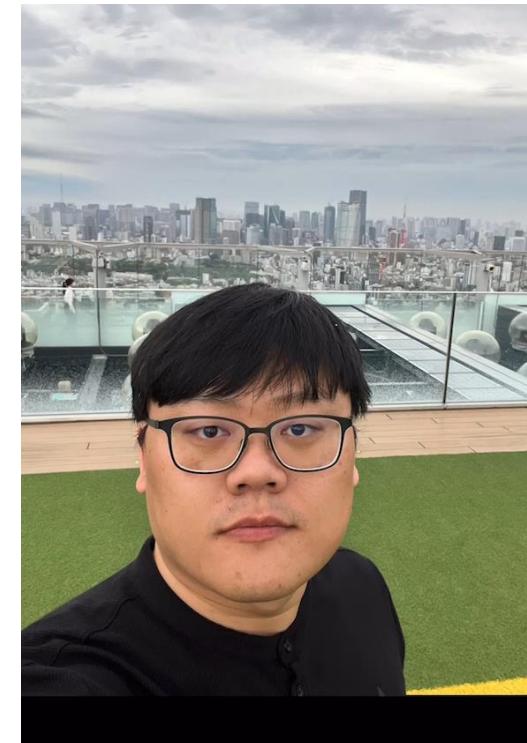
**3. 線性回歸與邏輯迴歸的差異（20分）**
請解釋線性回歸（Linear Regression）和邏輯迴歸（Logistic Regression）之間的主要差異，並說明每個方法適用於 ↓ 類型的問題。

# 生成式人工智慧(GAI)

人工智慧(AI):利用電腦模仿人類智慧的技術

機器學習(ML):利用數據進行分析與預測的技術

深度學習(DL):一種機器學習方法。利用多層次的類神經機器學習方法進行數據的特徵學習以辨識多特徵的物件、語言

生成式AI(GAI):基於特殊深度學習算法(ex: GAN, Transfomer Model)，以產生新特徵的內容、圖像、語言等。ChatGPT的預訓練大規模語言模型是目前主流的生成式AI作法

自然語言(NLP):人工智慧與語言學的分支，專注於人類語言的認知、理解、生成等

AI
ML
DL
GAI
NLP

Adobe just announced Project Perfect Blend!

*Mind blowing* 🤯
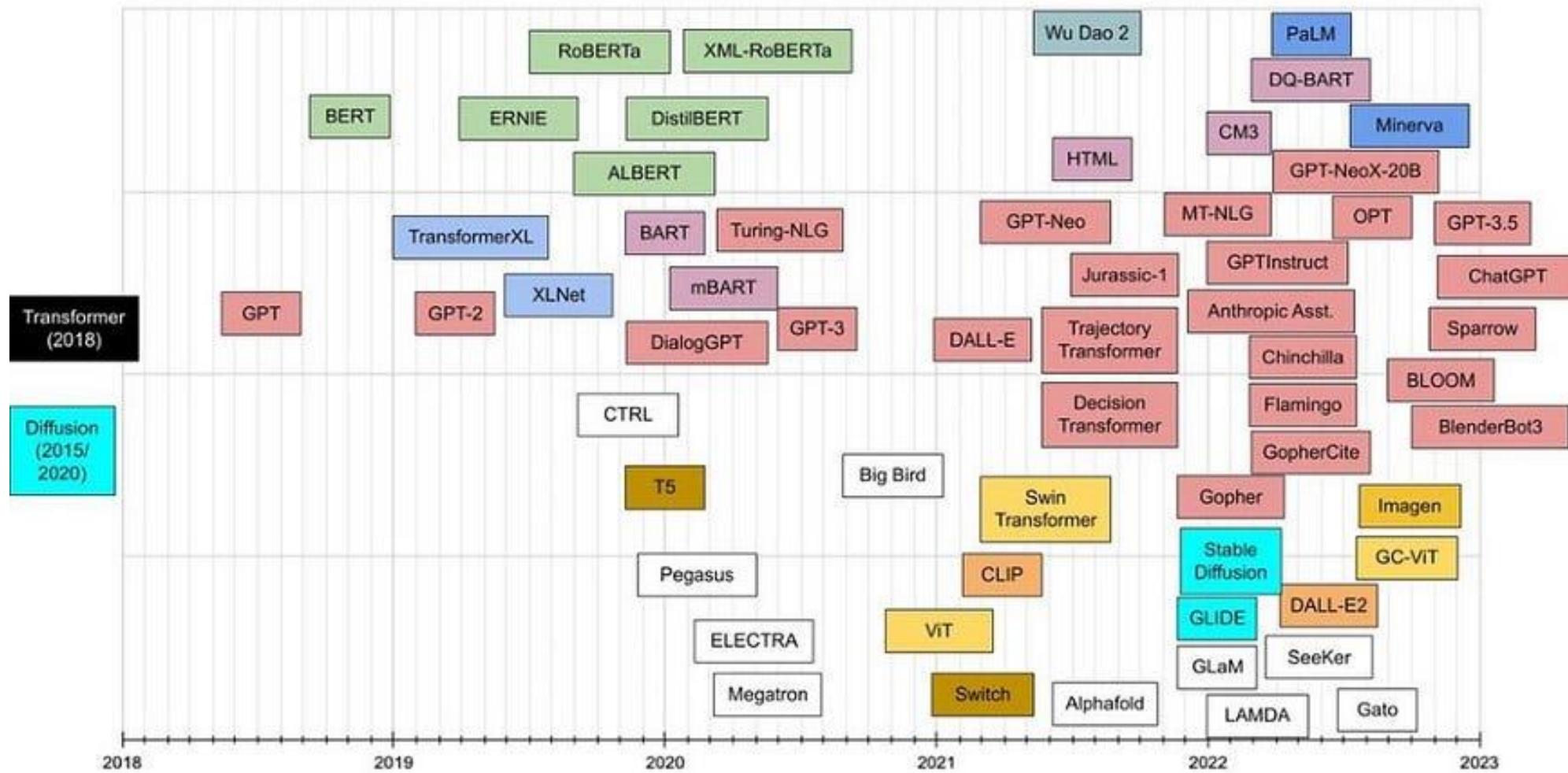
GAI ⟶ 文字處理和生成: LLMs相關
圖片生成: GAN/Diffusion model

圖、生成式AI與人工智慧(資料來源: Gartner, Accenture)
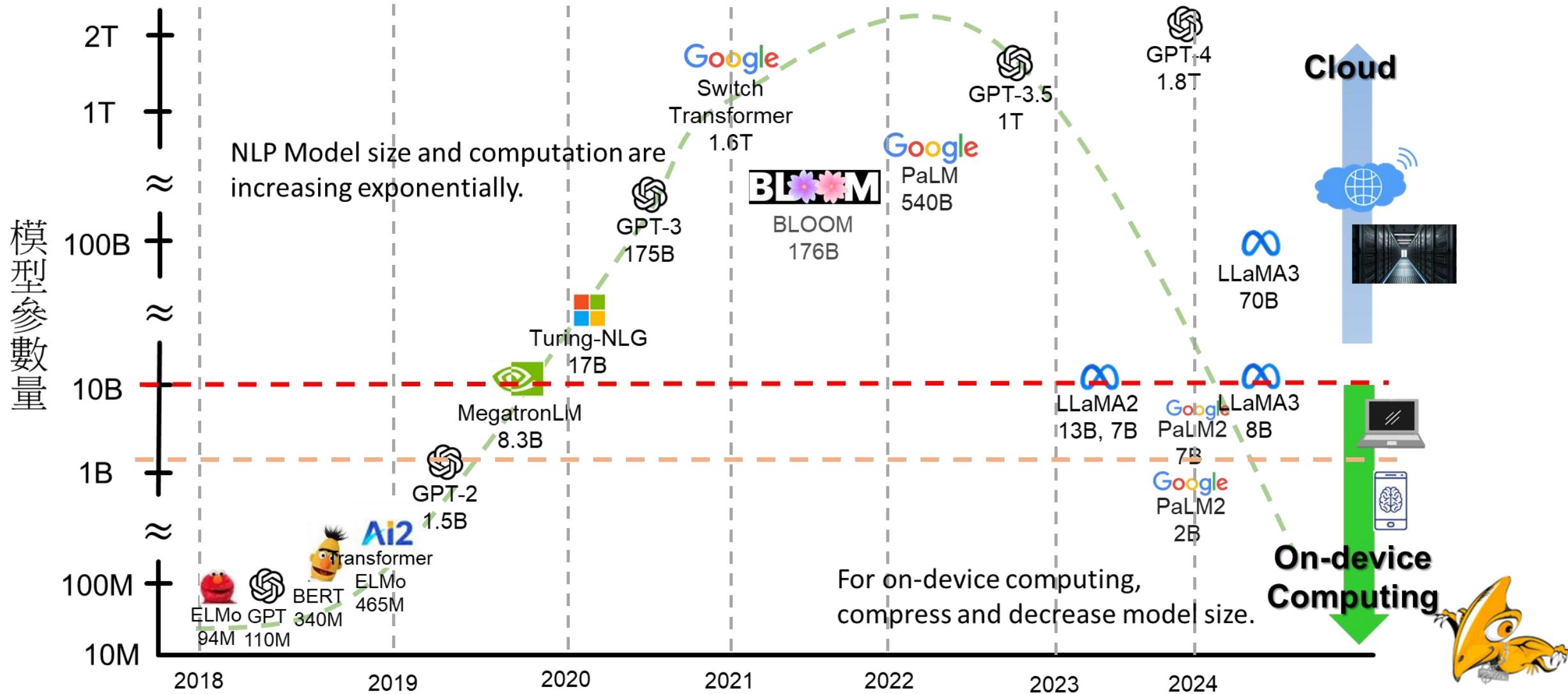
# 語言模型的發展歷程
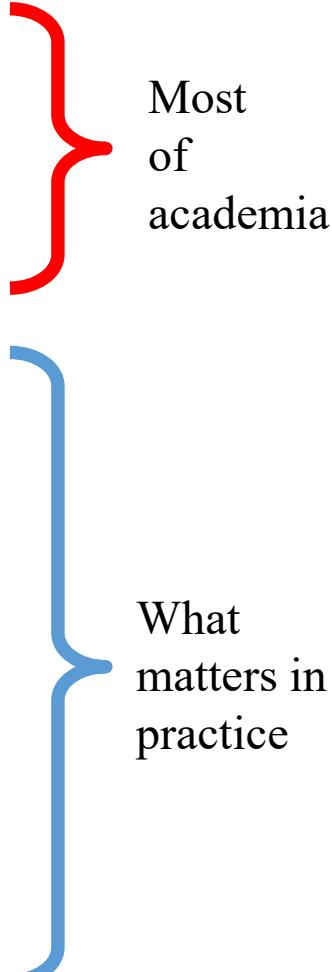
# 參數量和模型發展

參數量到一個程度後，生成AI已經有比較好的效果，模型發展往輕量化處理，讓AI可以在device上實現。

NLP Model size and computation are increasing exponentially.

For on-device computing, compress and decrease model size.

模型參數量

- 2T
- 1T
- ≈
- 100B
- ≈
- 10B
- 1B
- ≈
- 100M
- 10M

2018  2019  2020  2021  2022  2023  2024

**Cloud**

**On-device Computing**

- ELMo 94M
- GPT 110M
- BERT 340M
- Transformer ELMo 465M
- GPT-2 1.5B
- MegatronLM 8.3B
- Turing-NLG 17B
- GPT-3 175B
- Switch Transformer 1.6T
- BLOOM 176B
- PaLM 540B
- GPT-3.5 1T
- GPT-4 1.8T
- LLaMA2 13B, 7B
- PaLM2 7B
- PaLM2 2B
- LLaMA3 8B
- LLaMA3 70B

# AI怎麼運作

1. Model architecture

2. Training algorithm/skill:
      loss function, optimizer,... etc

Most
of
academia

3. Data

4. Evaluation

5. System:
      server architecture
      edge device
      operations,...
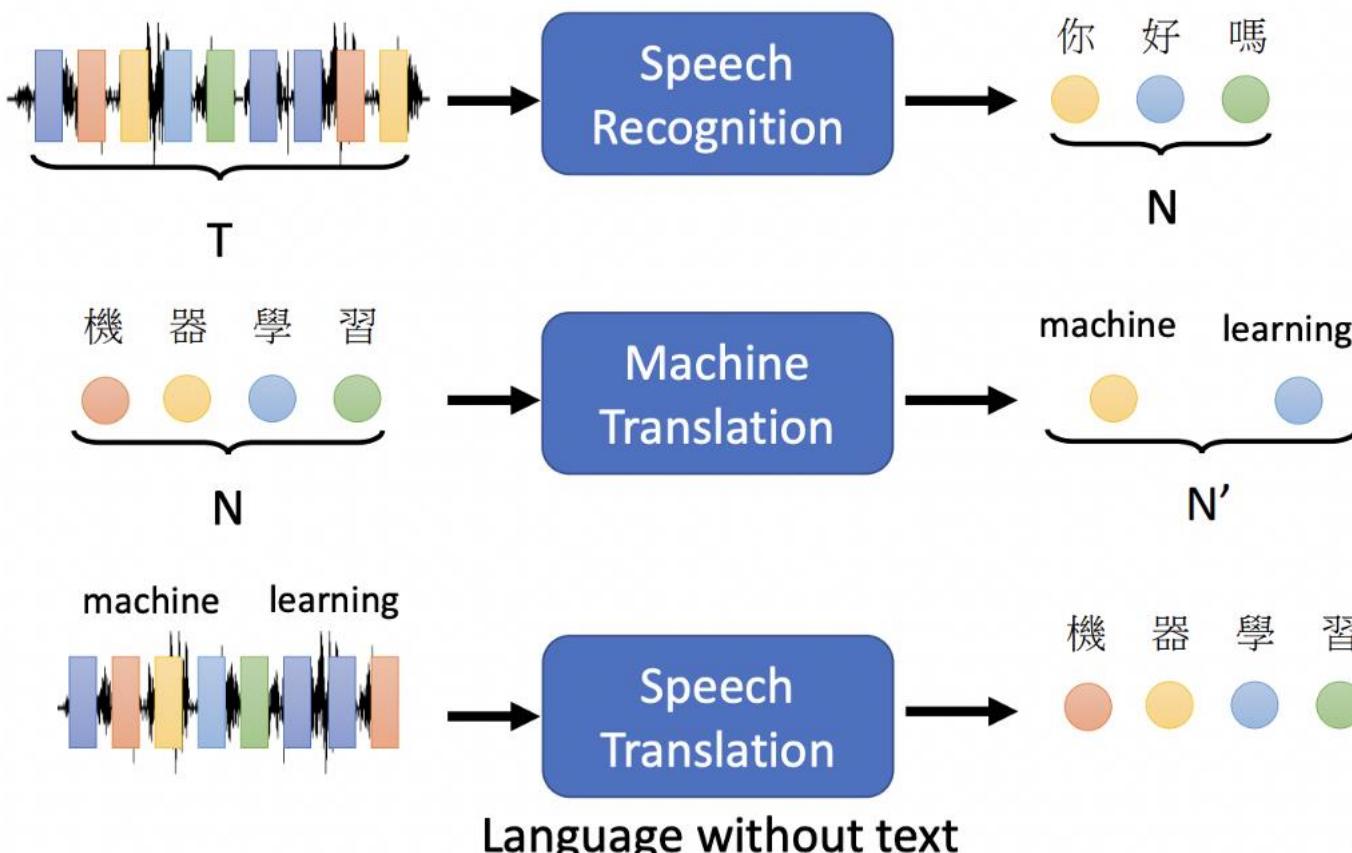
What
matters in
practice

# Sequence-to-sequence
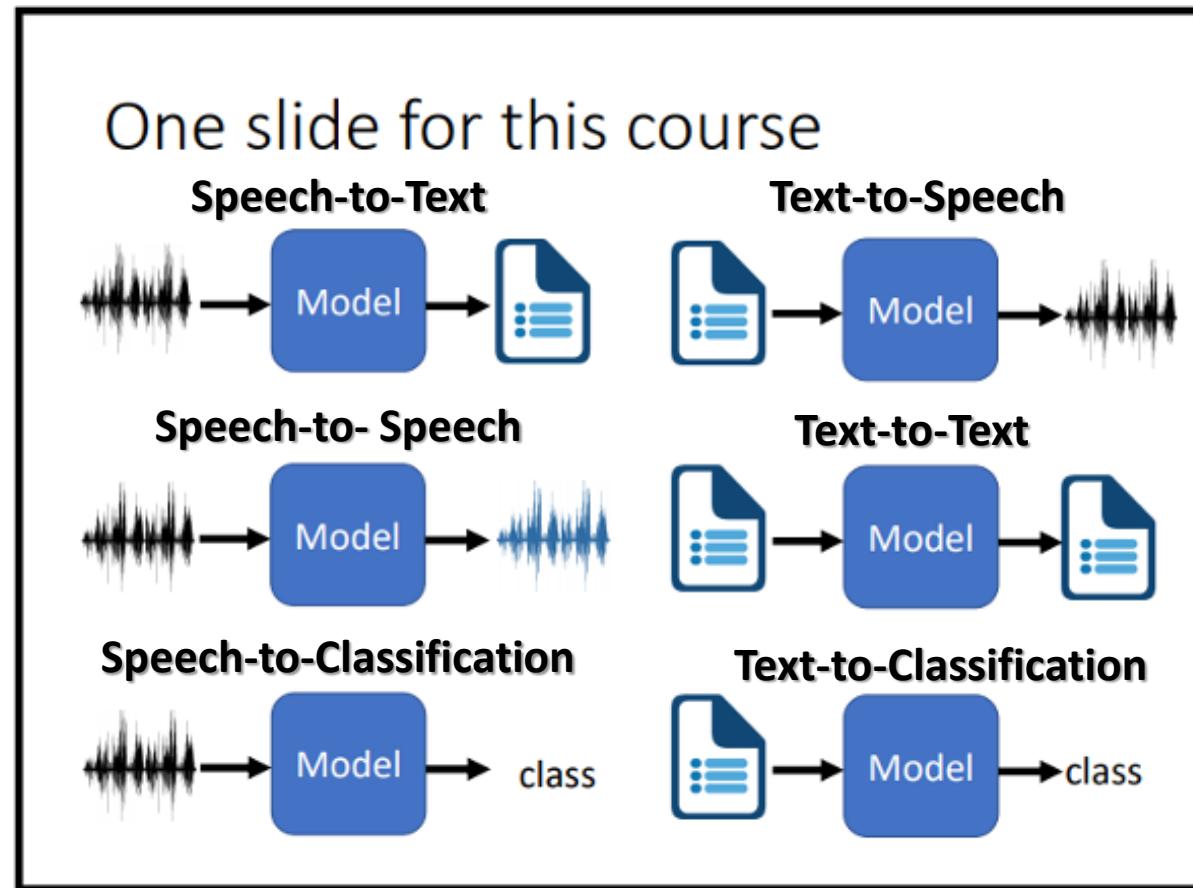


圖片 : 李宏毅老師

# Deep Learning for Human Language Processing
## 深度學習與人類語言處理



One slide for this course

Speech-to-Text

Text-to-Speech

Speech-to- Speech

Text-to-Text

Speech-to-Classification

Text-to-Classification

Source webpage: https://speech.ee.ntu.edu.tw/~hylee/dlhlp/2020-spring.html
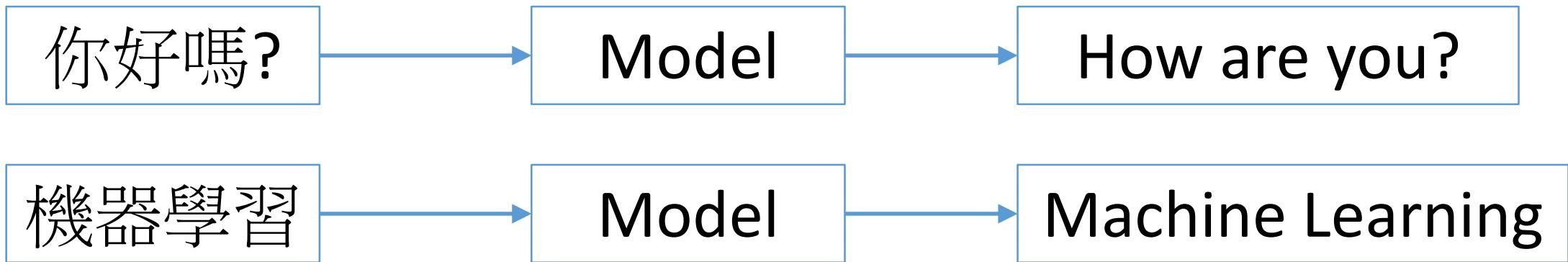
圖片：李宏毅老師

# Sequence-to-sequence
# Text-to-Text: 中英文翻譯

| 你好嗎? | → | Model | → | How are you? |
|---|---|---|---|---|

| 機器學習 | → | Model | → | Machine Learning |
|---|---|---|---|---|

電腦怎麼知道
機器學習→Machine Learning
你好嗎? →How are you?

# Sequence-to-sequence
# Text-to-Text: 中英文翻譯

電腦怎麼知道
機器學習→Machine Learning
你好嗎? →How are you?

給電腦一堆配對好的資料，讓模
型去學中文和英文之間的關聯性。

Example/dataset.xlsx

| | A | B |
|---|---|---|
| 1 | english | chinese |
| 2 | Hi. | 嗨。 |
| 3 | Hi. | 你好。 |
| 4 | Run. | 你用跑的。 |
| 5 | Wait! | 等等！ |
| 6 | Wait! | 等一下！ |
| 7 | Begin. | 开始！ |
| 8 | Begin | 开始 |
| 9 | Hello! | 你好。 |
| 10 | I try. | 我试试。 |
| 11 | I won! | 我赢了。 |
| 12 | Oh no! | 不会吧。 |
| 13 | Cheers! | 乾杯! |
| 14 | Got it? | 你懂了吗？ |
| 15 | He ran. | 他跑了。 |
| 16 | Hop in. | 跳进来。 |

# Tokenization/Tokenizer

| 你好嗎? | → | Model | → | How are you? |
|---|---|---|---|---|

・ Why do we need Tokenization?

你好嗎? → "你"、"好"、"嗎"、"?"

How are you? → " How "、" are "、" you"、"?"

**1. 基於詞的(Word-based):**

我是黃志勝
→ "我"、"是"、"黃"、"志"、"勝"
→ "我是"、"黃志勝"
→ "我"、"是"、"黃"、"志勝"

**2. 基於字符(Character-based):**

How are you? → "H "、"o"、"w"、"a "、"r"、"e"、"y"、"o"、"u"、"?"

# Tokenization/Tokenizer

你好嗎? **?** → Model **?** → How are you?
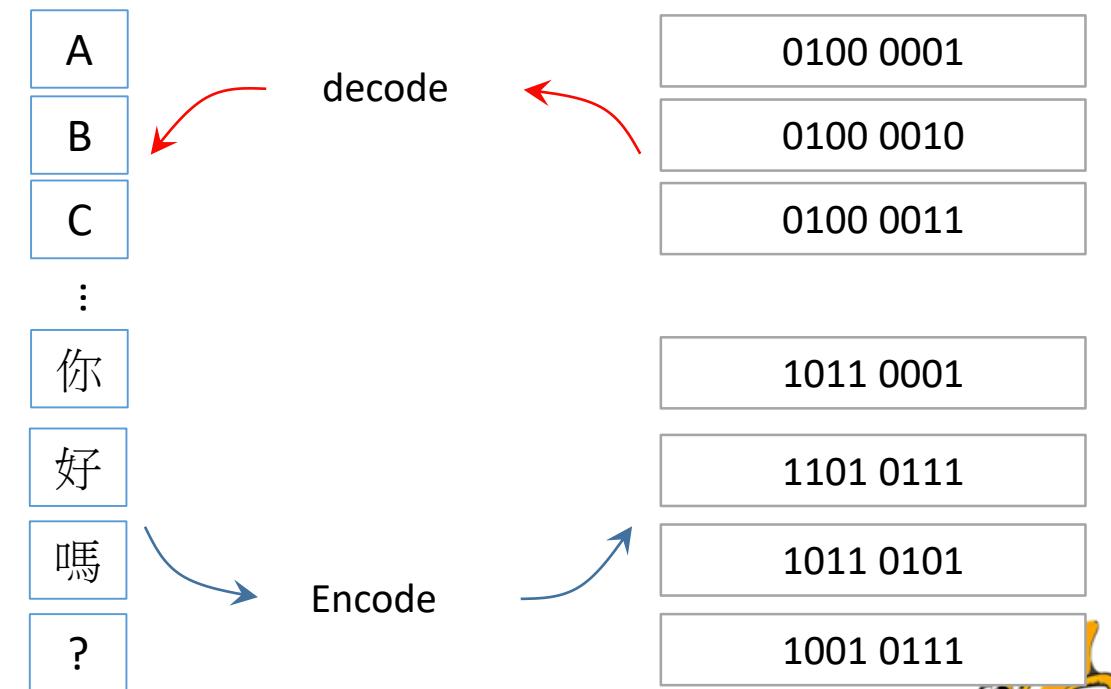
編碼:
ASCII
Big5
Unicode
Base64

電腦看得懂文字(中文/英文/日文/...)**?**

編碼: **ASCII**

| 二進制 | 十進制 | 十六進制 | 圖形 | 二進制 | 十進制 | 十六進制 | 圖形 | 二進制 | 十進制 | 十六進制 | 圖形 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0010 0000 | 32 | 20 | (space) | 0100 0000 | 64 | 40 | @ | 0110 0000 | 96 | 60 | ` |
| 0010 0001 | 33 | 21 | ! | 0100 0001 | 65 | 41 | A | 0110 0001 | 97 | 61 | a |
| 0010 0010 | 34 | 22 | " | 0100 0010 | 66 | 42 | B | 0110 0010 | 98 | 62 | b |
| 0010 0011 | 35 | 23 | # | 0100 0011 | 67 | 43 | C | 0110 0011 | 99 | 63 | c |
| 0010 0100 | 36 | 24 | $ | 0100 0100 | 68 | 44 | D | 0110 0100 | 100 | 64 | d |
| 0010 0101 | 37 | 25 | % | 0100 0101 | 69 | 45 | E | 0110 0101 | 101 | 65 | e |
| 0010 0110 | 38 | 26 | & | 0100 0110 | 70 | 46 | F | 0110 0110 | 102 | 66 | f |
| 0010 0111 | 39 | 27 | ' | 0100 0111 | 71 | 47 | G | 0110 0111 | 103 | 67 | g |
| 0010 1000 | 40 | 28 | ( | 0100 1000 | 72 | 48 | H | 0110 1000 | 104 | 68 | h |

A
B
C
:
你
好
嗎
?

decode

Encode

0100 0001
0100 0010
0100 0011

1011 0001
1101 0111
1011 0101
1001 0111
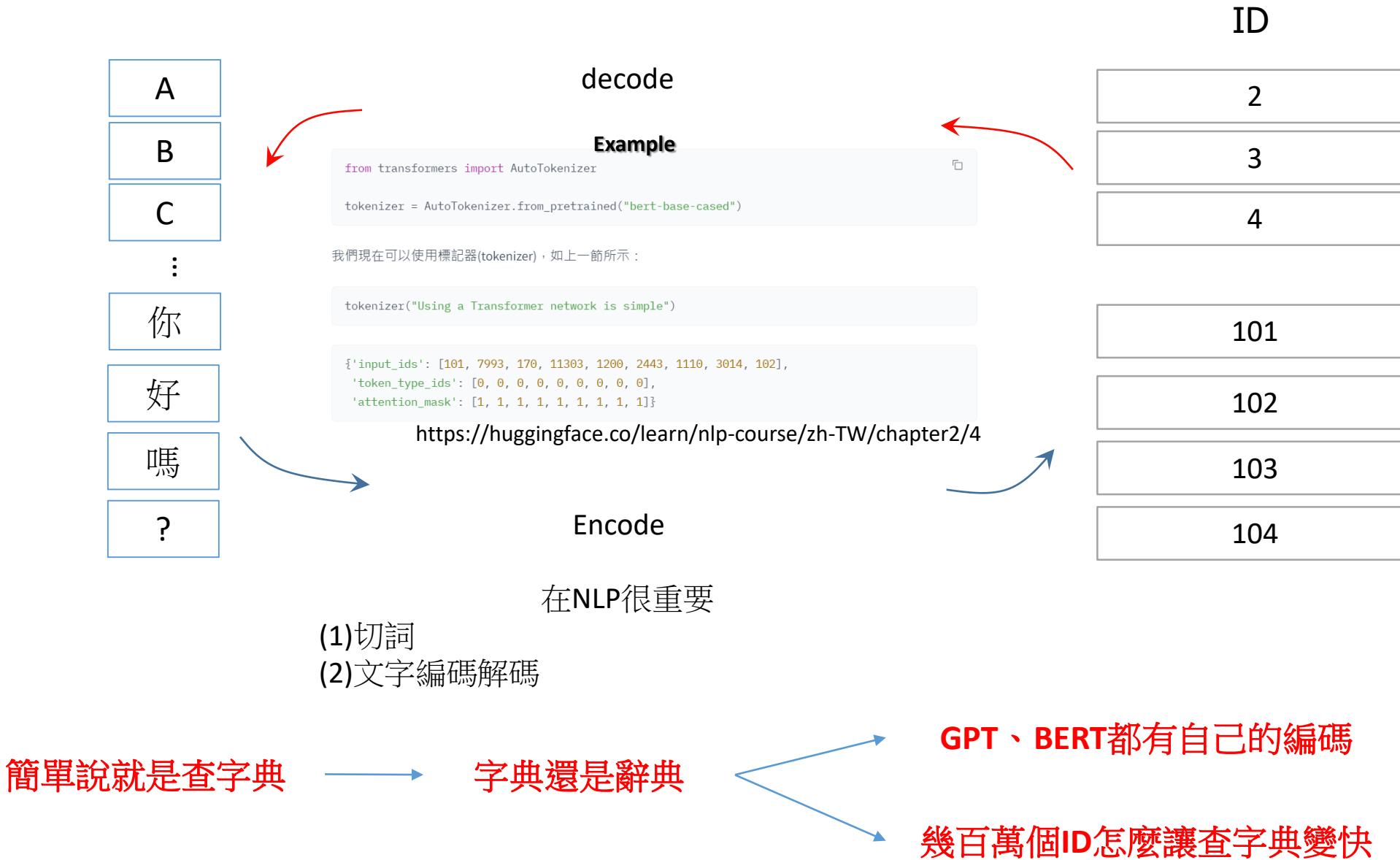
Note:中文編碼是我亂打的

# **Tokenization/Tokenizer**

ID

| | |
|---|---|
| A | 2 |
| B | 3 |
| C | 4 |
| ⋮ | |
| 你 | 101 |
| 好 | 102 |
| 嗎 | 103 |
| ? | 104 |

decode

### Example

```
from transformers import AutoTokenizer

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

我們現在可以使用標記器(tokenizer)，如上一節所示：

```
tokenizer("Using a Transformer network is simple")
```

```
{'input_ids': [101, 7993, 170, 11303, 1200, 2443, 1110, 3014, 102],
 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0],
 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

https://huggingface.co/learn/nlp-course/zh-TW/chapter2/4

Encode

在NLP很重要

(1)切詞
(2)文字編碼解碼

簡單說就是查字典 → 字典還是辭典

GPT、BERT都有自己的編碼

幾百萬個ID怎麼讓查字典變快

# Tokenization/Tokenizer

Example/dataset.xlsx

| | A | B |
|---|---|---|
| 1 | english | chinese |
| 2 | Hi. | 嗨。 |
| 3 | Hi. | 你好。 |
| 4 | Run. | 你用跑的。 |
| 5 | Wait! | 等等！ |
| 6 | Wait! | 等一下！ |
| 7 | Begin. | 开始！ |
| 8 | Begin | 开始 |
| 9 | Hello! | 你好。 |
| 10 | I try. | 我试试。 |
| 11 | I won! | 我赢了。 |
| 12 | Oh no! | 不会吧。 |
| 13 | Cheers! | 乾杯! |
| 14 | Got it? | 你懂了吗？ |
| 15 | He ran. | 他跑了。 |
| 16 | Hop in. | 跳进来。 |

Word Embedding:
將『字詞/句子/文件』轉換成『向量』形式

英文有自己的字典　　　　中文有自己的字典
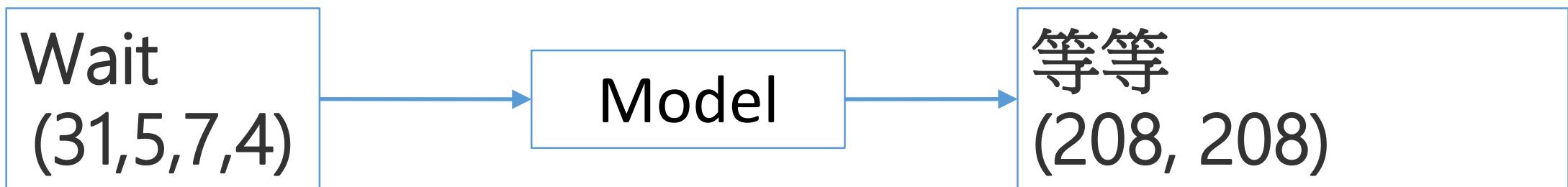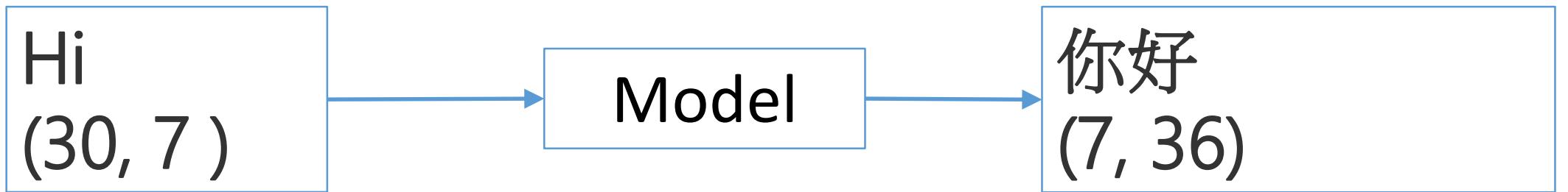
Hi ⟶ [30, 7]　　　　你好 ⟶ [7, 36]

Wait ⟶ [31, 5, 7, 4]　　　等等 ⟶ [208, 208]

# Sequence-to-sequence
# **Text-to-Text:** 中英文翻譯

| Hi (30, 7 ) | → Model → | 你好 (7, 36) |

| Wait (31,5,7,4) | → Model → | 等等 (208, 208) |

# Sequence-to-sequence

Hi
(30, 7 )

Model

你好
(7, 36)

Output

Encoder → Decoder

Input

Output

Encoder → Decoder

Input

| h1 | h2 | ht | 你 | 好 | ... | <EOS> |

Encoder

Decoder

| H | i | ... | <EOS> | <BOS> | 你 | ... | <EOS> |
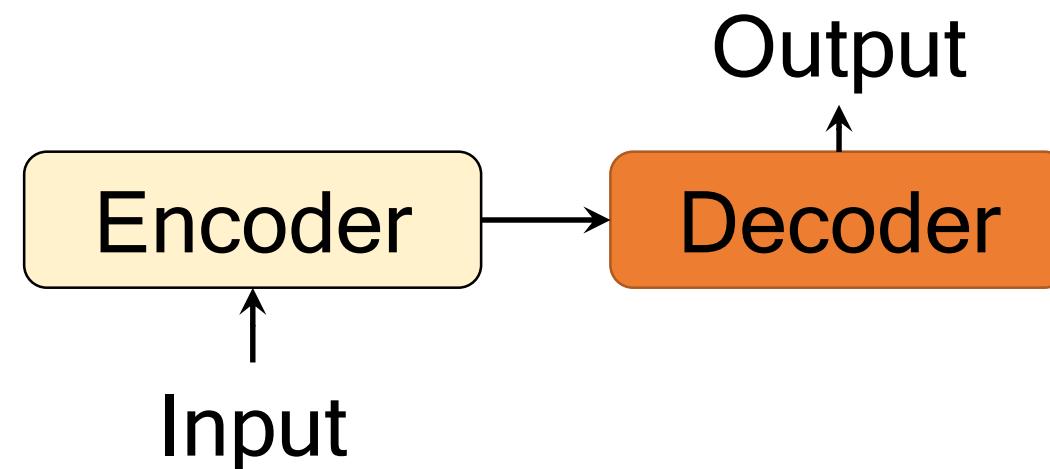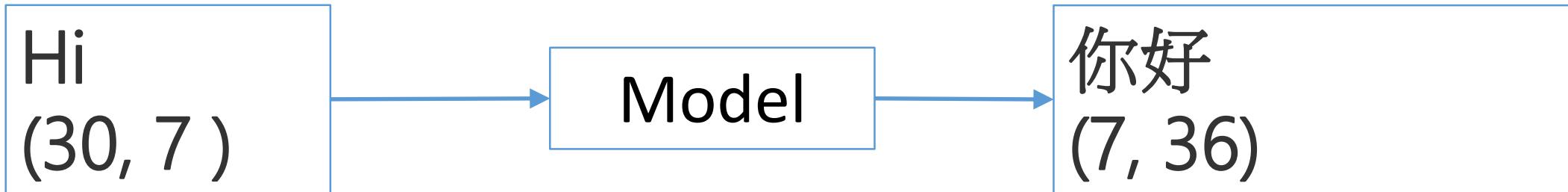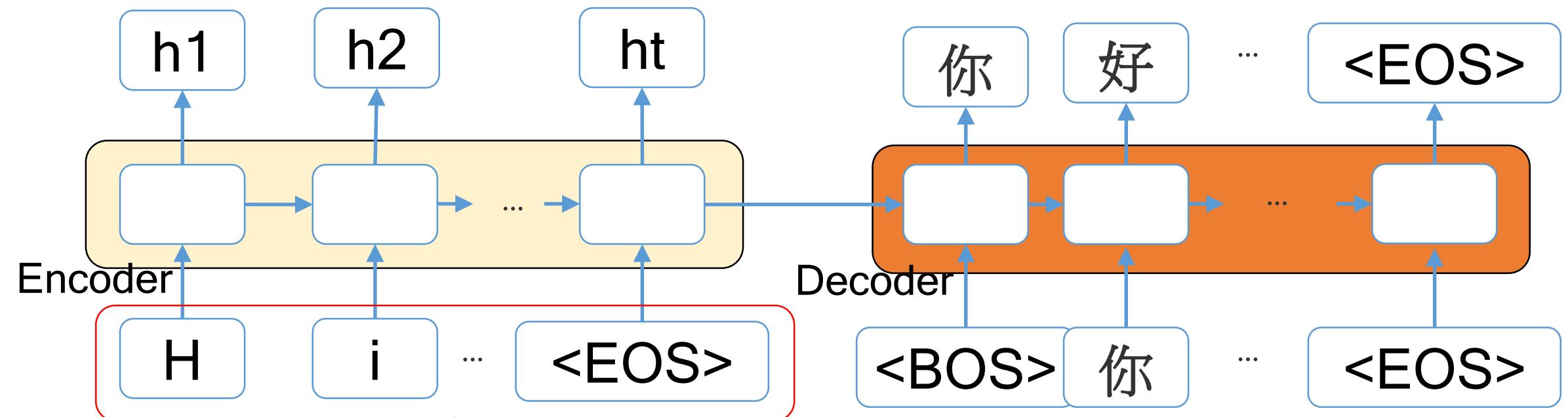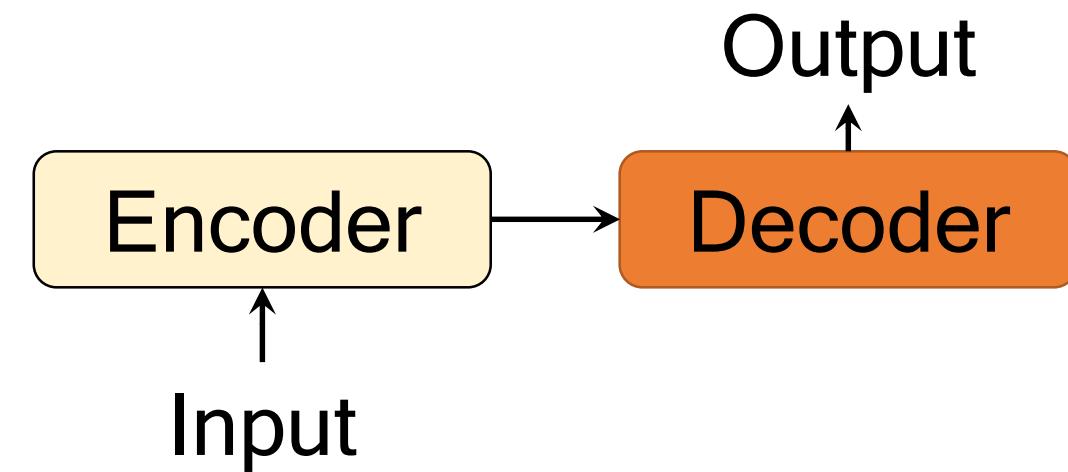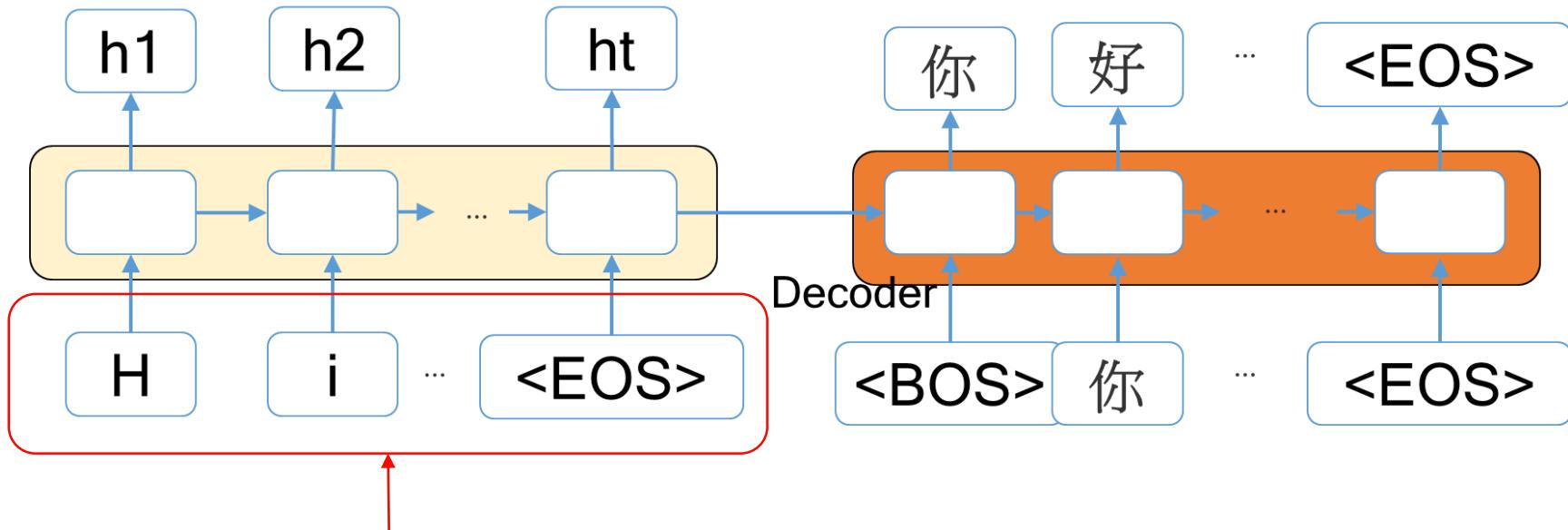
EOS: End of Sentence (EOT: End of Token)
BOS: Begin of Sentence

Token長度要多長?

Token長度要多長?

The previous example runs until you hit the model's token limit. With each question asked, and answer received, the `messages` list grows in size. The token limit for `gpt-35-turbo` is 4,096 tokens. The token limits for `gpt-4` and `gpt-4-32k` are 8,192 and 32,768, respectively. These limits include the token count from both the message list sent and the model response. The number of tokens in the messages list combined with the value of the `max_tokens` parameter must stay under these limits or you receive an error.
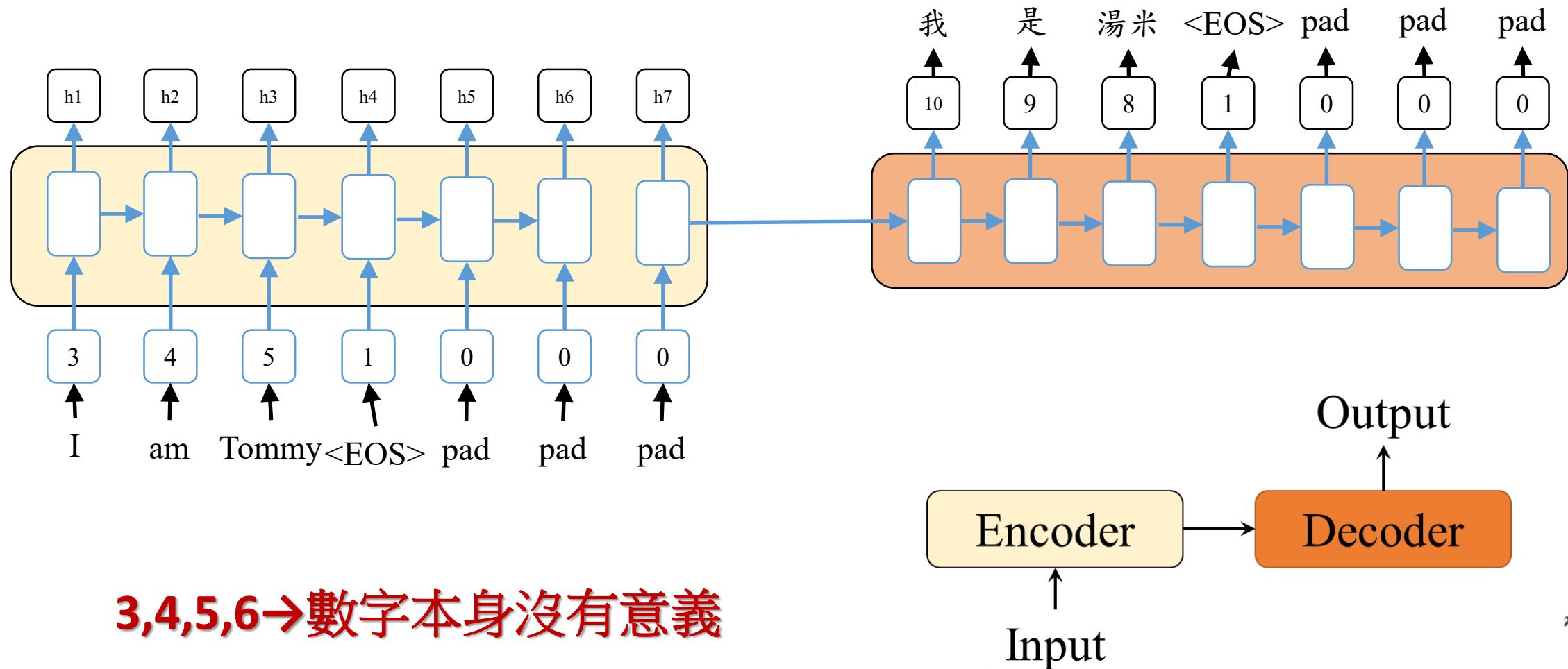
https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/chatgpt?tabs=python-new

# Tokenization

| 句子 | 切字 | 處理後的編碼 | 中文 | 切字 | 處理後的編碼 |
|---|---|---|---|---|---|
| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
| I am good. | 'I', 'am', 'good', '.' | [3, 4, 6, 10, 1, 0, 0] | 我很好 | '我', '很好' | [10, 7, 1, 0, 0, 0, 0] |
| good to see you. | 'good', 'to', 'see', 'you', '.' | [6, 7, 8, 9, 10, 1, 0] | 很高興見到你 | '很', '高興', '見到', '你' | [6, 5, 4, 3, 1, 0, 0 ] |

| 英文字典 | 編碼 |
|---|---|
| I | 3 |
| Am | 4 |
| Tommy | 5 |
| Good | 6 |
| To | 7 |
| See | 8 |
| You | 9 |
| . | 10 |
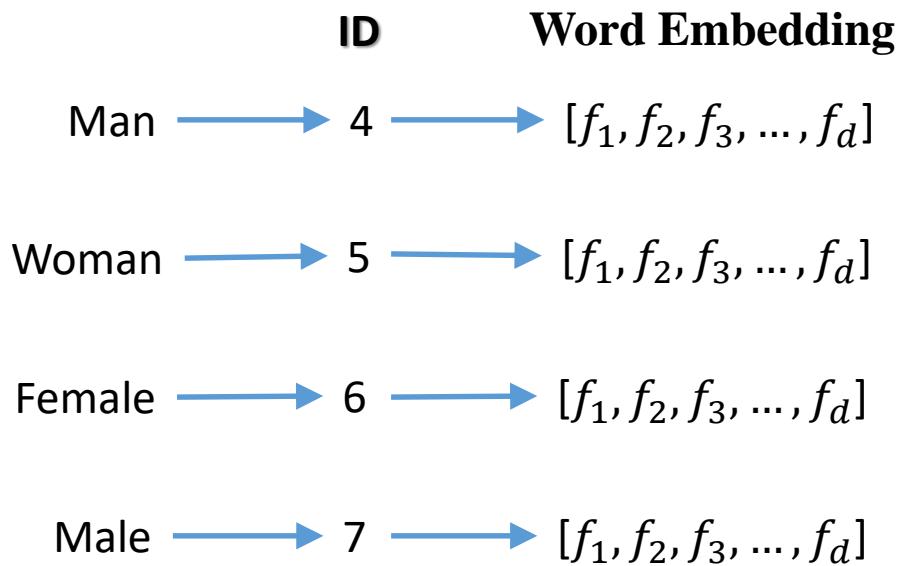
Output

Encoder → Decoder

Input

<EOS> : 1
Pad : 0
<BOS> : 2

| 中文字典 | 編碼 |
|---|---|
| 我 | 10 |
| 是 | 9 |
| 湯米 | 8 |
| 很好 | 7 |
| 很 | 6 |
| 高興 | 5 |
| 見到 | 4 |
| 你 | 3 |

# Tokenization: Word Embedding

| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
|---|---|---|---|---|---|



**3,4,5,6→數字本身沒有意義**

# Tokenization: Word Embedding

| | ID | Word Embedding |
|---|---|---|
| Man | 4 | $[f_1, f_2, f_3, \ldots, f_d]$ |
| Woman | 5 | $[f_1, f_2, f_3, \ldots, f_d]$ |
| Female | 6 | $[f_1, f_2, f_3, \ldots, f_d]$ |
| Male | 7 | $[f_1, f_2, f_3, \ldots, f_d]$ |

- 用更多維度來表示每一個字
- 也可以在訓練模型的時候順便學習

Google: Word2vec
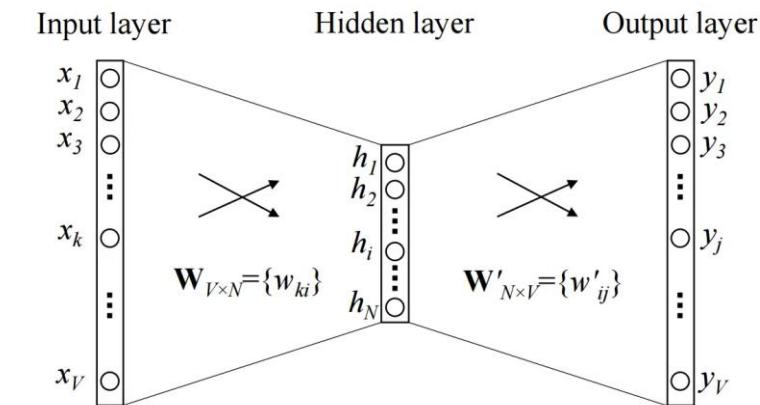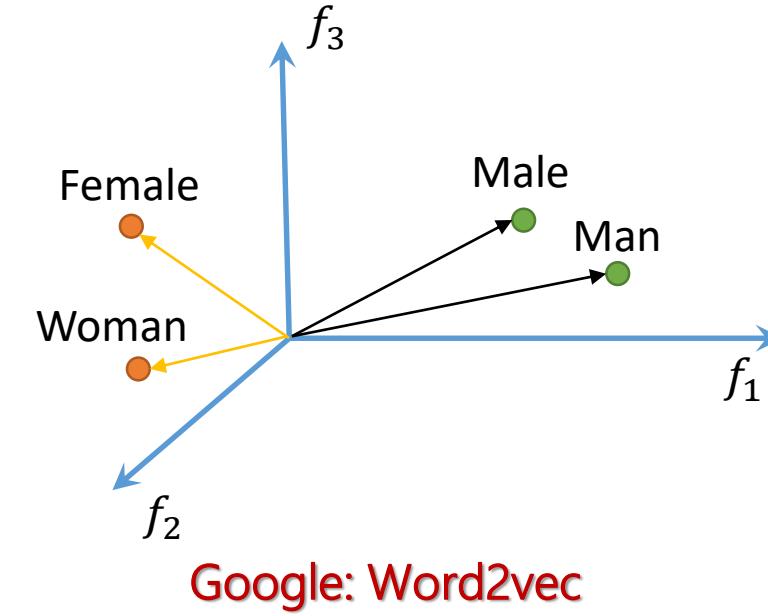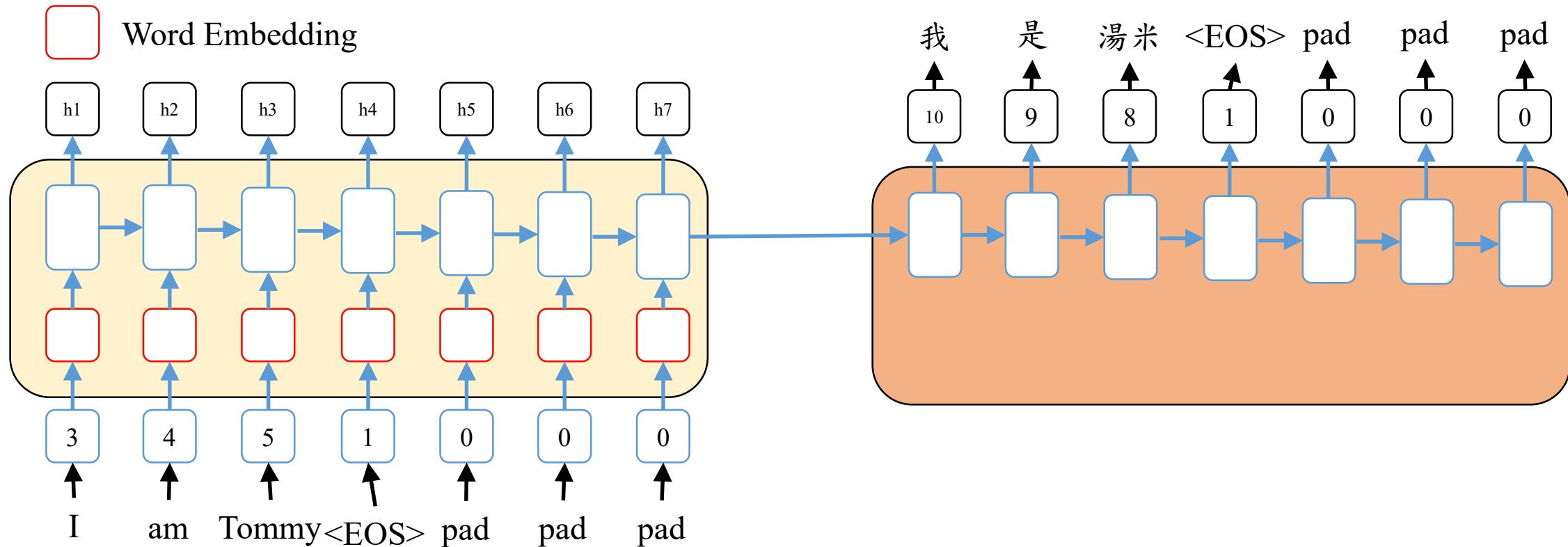
Figure 1: A simple CBOW model with only one word in the context
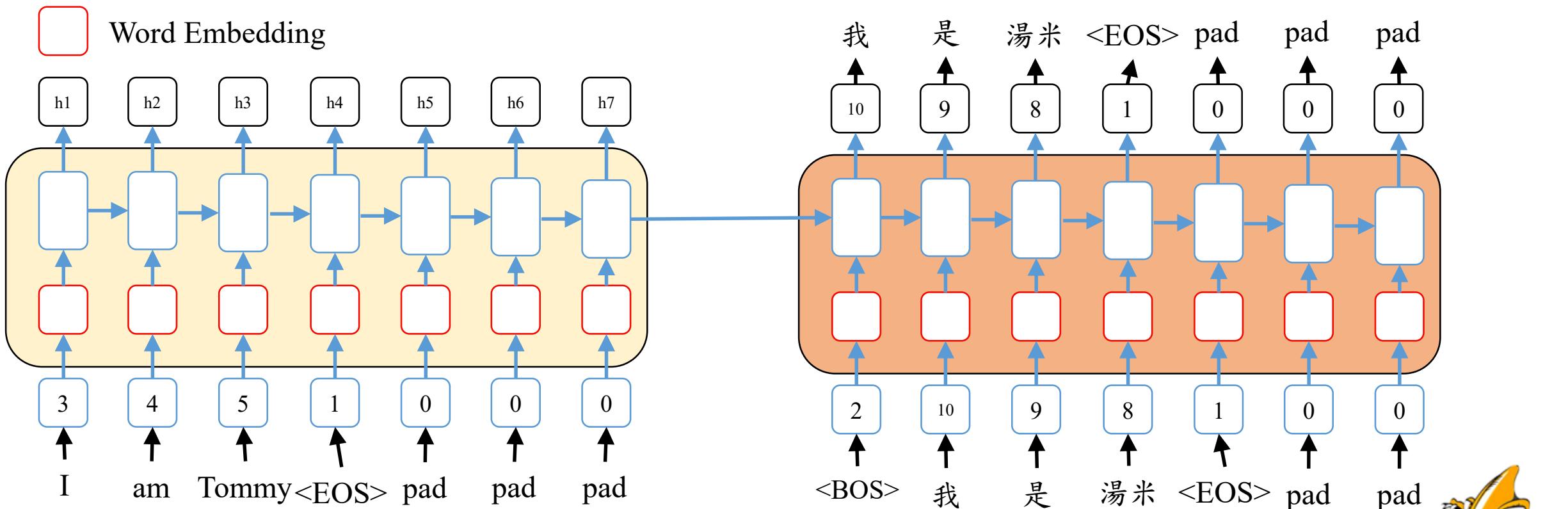
# Tokenization: Word Embedding

| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
|---|---|---|---|---|---|

# 範例-I am Tommy
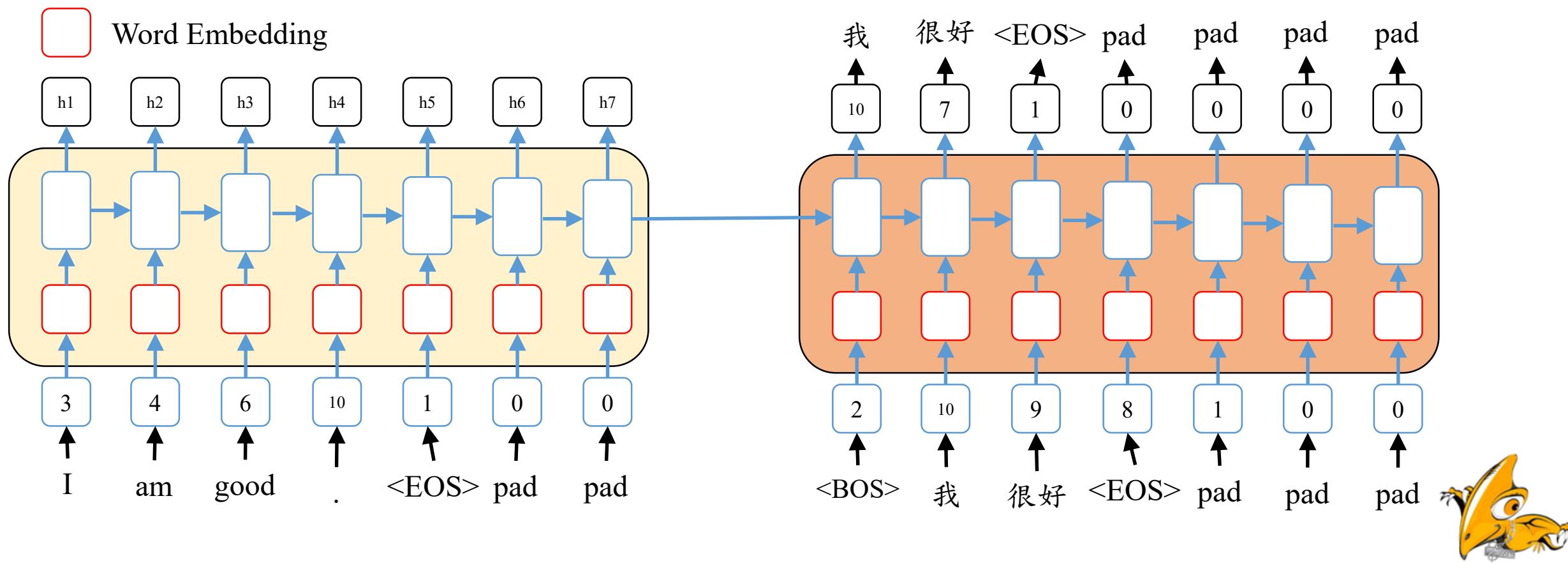
| 句子 | 切字 | 處理後的編碼 | 中文 | 切字 | 處理後的編碼 |
|------|------|------------|------|------|------------|
| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
| I am good. | 'I', 'am', 'good', '.' | [3, 4, 6, 10, 1, 0, 0] | 我很好 | '我', '很好' | [10, 7, 1, 0, 0, 0, 0] |
| good to see you. | 'good', 'to', 'see', 'you', '.' | [6, 7, 8, 9, 10, 1, 0] | 很高興見到你 | '很', '高興', '見到', '你' | [6, 5, 4, 3, 1, 0, 0 ] |

# 範例-I am good.

| 句子 | 切字 | 處理後的編碼 | 中文 | 切字 | 處理後的編碼 |
|------|------|------|------|------|------|
| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
| I am good. | 'I', 'am', 'good', '.' | [3, 4, 6, 10, 1, 0, 0] | 我很好 | '我', '很好' | [10, 7, 1, 0, 0, 0, 0] |
| good to see you. | 'good', 'to', 'see', 'you', '.' | [6, 7, 8, 9, 10, 1, 0] | 很高興見到你 | '很', '高興', '見到', '你' | [6, 5, 4, 3, 1, 0, 0 ] |



Word Embedding

| h1 | h2 | h3 | h4 | h5 | h6 | h7 |

Encoder inputs: 3 | 4 | 6 | 10 | 1 | 0 | 0

I | am | good | . | <EOS> | pad | pad

Decoder outputs: 我 | 很好 | <EOS> | pad | pad | pad | pad

| 10 | 7 | 1 | 0 | 0 | 0 | 0 |

Decoder inputs: 2 | 10 | 9 | 8 | 1 | 0 | 0

<BOS> | 我 | 很好 | <EOS> | pad | pad | pad
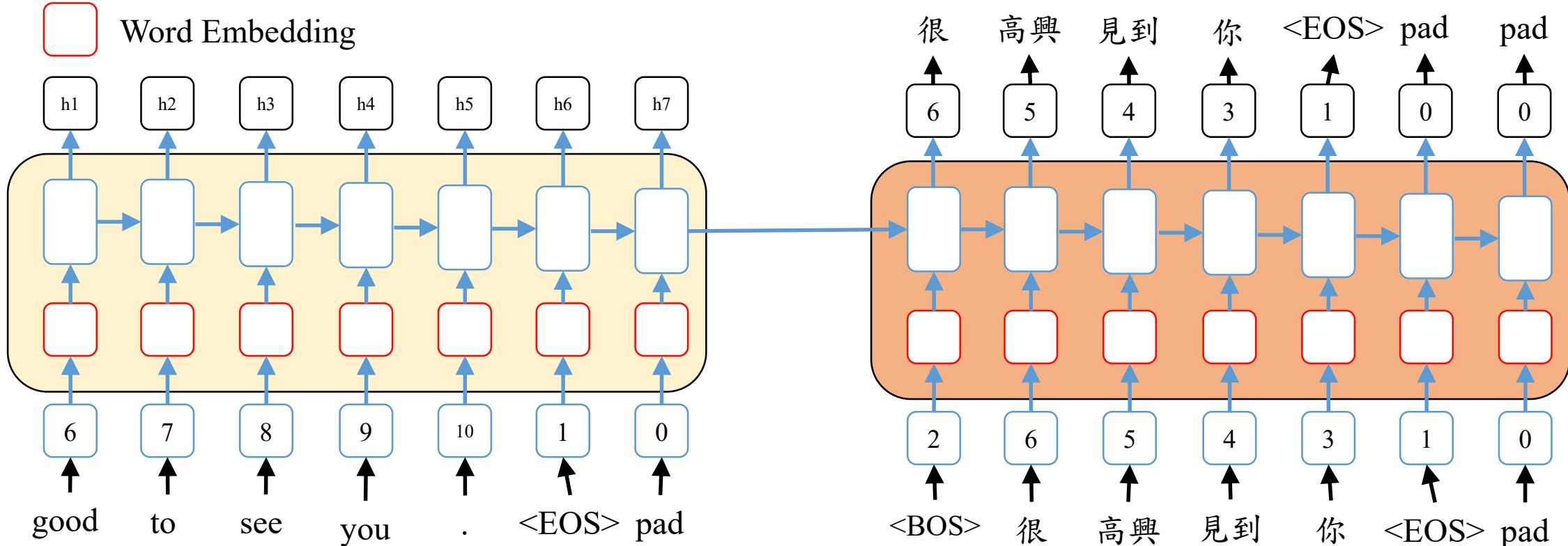
# 範例-good to see you.

| 句子 | 切字 | 處理後的編碼 | 中文 | 切字 | 處理後的編碼 |
|---|---|---|---|---|---|
| I am Tommy | 'I', 'am', 'Tommy' | [3, 4, 5, 1, 0, 0, 0] | 我是湯米 | '我', '是', '湯米' | [10, 9, 8, 1, 0, 0, 0 ] |
| I am good. | 'I', 'am', 'good', '.' | [3, 4, 6, 10, 1, 0, 0] | 我很好 | '我', '很好' | [10, 7, 1, 0, 0, 0, 0] |
| good to see you. | 'good', 'to', 'see', 'you', '.' | [6, 7, 8, 9, 10, 1, 0] | 很高興見到你 | '很', '高興', '見到', '你' | [6, 5, 4, 3, 1, 0, 0 ] |

# Tokenizer: Byte Pair Encoding

| Word | 字典 | Tokenization ID |
|------|------|-----------------|
| **Book** ⟶ | **Book** ⟶ | **101** |
| **is** ⟶ | **is** ⟶ | **2505** |
| **good** ⟶ | **good** ⟶ | **500** |
| **booook** ⟶ | **Unknown** ⟶ | **?** |
| **tarsformer** ⟶ | **Unknown** ⟶ | **?** |

正常

打錯

**Word最小單位?**    Out-of-vocabulary (OOV)

# Token Definition: Byte Pair Encoding
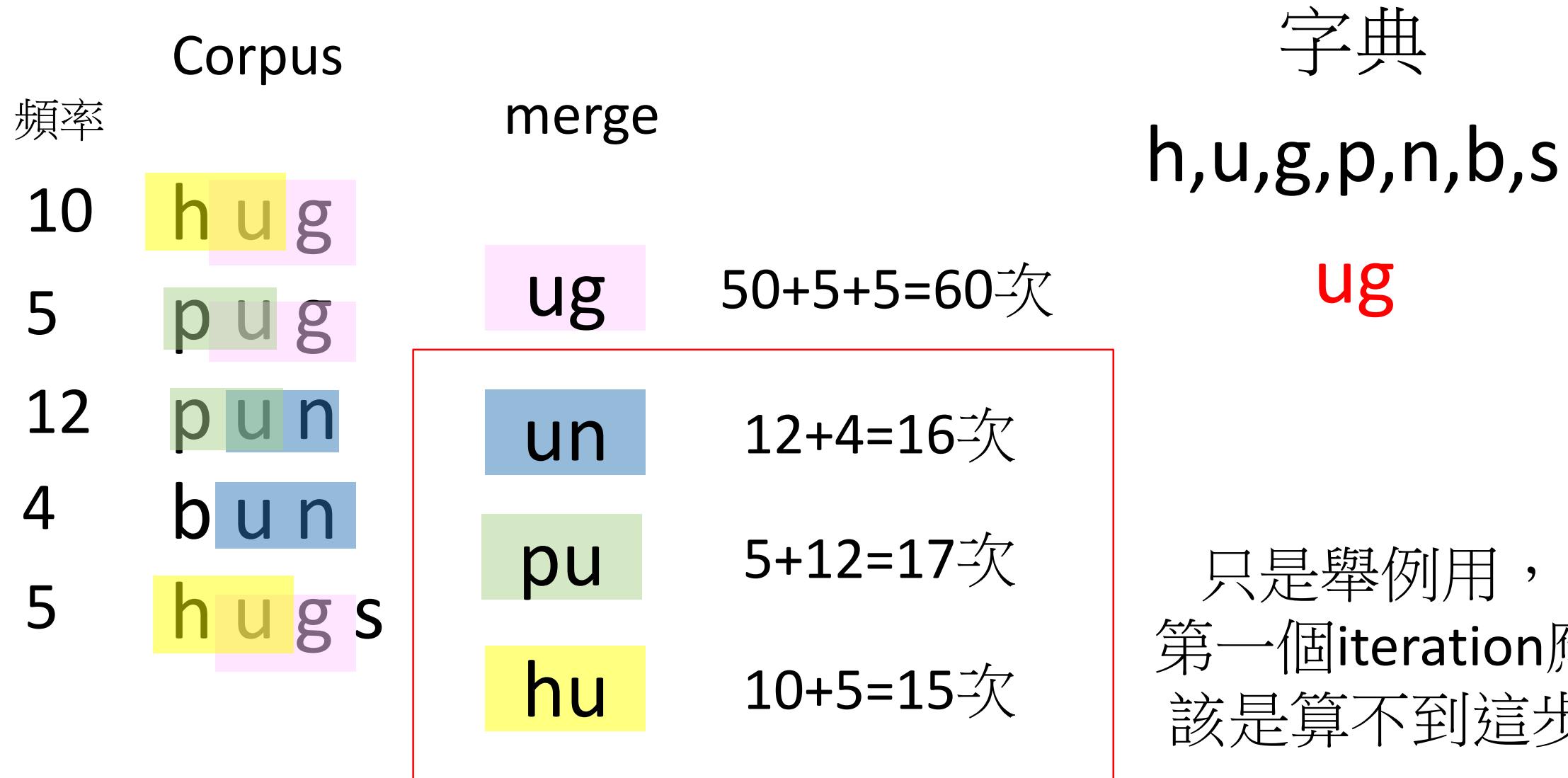
- **Character** : 前面介紹過，就切到最小單位
  缺點: 語意很難被認清楚
  pineapple, pine apple

- **Subword**: 部分的字，介於Word和Character

**Byte Pair Encoding (BPE) → subword vocabulary**

The original version of the algorithm focused on compression. It replaces the highest-frequency pair of bytes with a new byte that was not contained in the initial dataset. A lookup table of the replacements is required to rebuild the initial dataset.

# Tokenizer: Byte Pair Encoding

Corpus

字典

頻率

h,u,g,p,n,b,s

merge

| 頻率 | Corpus |
|------|--------|
| 10 | h u g |
| 5 | p u g |
| 12 | p u n |
| 4 | b u n |
| 5 | h u g s |

ug    50+5+5=60次

ug

un    12+4=16次

pu    5+12=17次

hu    10+5=15次

只是舉例用，
第一個iteration應
該是算不到這步

# Tokenizer: Byte Pair Encoding

Corpus

字典

h,u,g,p,n,b,s

ug

un

頻率

merge

10  h ug

un    12+4=16次

5  p ug

hug    10+5=15次

12  p u n

4  b u n

5  h ug s

# **Tokenizer: Byte Pair Encoding**

Corpus

字典

頻率

h,u,g,p,n,b,s

merge

10　h ug

5　p ug

ug

hug　10+5=15次

un

12　p un

4　b un

hug

5　h ug s

# Tokenizer: Byte Pair Encoding

Corpus

字典

| 頻率 | | merge | |
|---|---|---|---|
| 10 | hug | | h,u,g,p,n,b,s |
| 5 | p ug | pug 5次 | ug |
| 12 | p un | pun 12次 | un |
| 4 | b un | bun 4次 | hug |
| 5 | hug s | hugs 5次 | pug |

pun

hugs

bun

# Tokenizer: Byte Pair Encoding

字典

頻率

| | |
|---|---|
| 10 | h u g |
| 5 | p u g |
| 12 | p u n |
| 4 | b u n |
| 5 | h u g s |

h,u,g,p,n,b,s

ug
un
hug
pug

pun
hugs
bun

新字

bug ⟶ b, ug

hung ⟶ h, un, g

# Tokenizer: Byte Pair Encoding

typos

**tokenizer** ⟶ **tokeizer**

**Pre-tokenizer**

Idea: tokens as common subsequences

Byte Pair Encoding (BPE). Train steps:

1. Take large corpus of text

2. Start with one token per character

3. Merge common pairs of tokens into a token

4. Repeat until desired vocab size or all merged

"t,o" → "to"
"to, k" → "tok"
"tok, e" → "toke"
"toke, n" → "token"

Real Tokenizer in GPT

Reference: Stanford CS229 I Machine Learning I Building Large Language Models (LLMs)

# BPE

- Morphemes: e.g. -est, -er, -ing, -ed

- A **morpheme** is any of the smallest meaningful constituents within a linguistic expression and particularly within a word. (WIKI)



https://platform.openai.com/tokenizer

# OpenAI Platform

老師教學生知識

GPT-4o & GPT-4o mini    GPT-3.5 & GPT-4    GPT-3 (Legacy)

老師教學生知識

Clear    Show example

**Tokens**    **Characters**
15    7

老師教學生知識

**Tokens**    **Characters**
15    7

[32003, 223, 30585, 104, 46763, 247, 27764, 116, 37955, 163, 253, 98, 164, 255, 246]

Clear    Show example

**Tokens**    **Characters**
5    7

老師教學生知識

**Tokenizing Chinese by Unicode encoding.**

Text    Token IDs

Text    Token IDs

Text    Token IDs
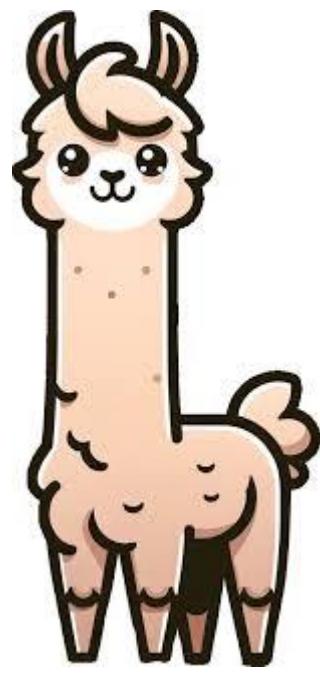
多語言難度高                                    包山包海

# The Llama 3 Herd of Models

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christoph Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone et al. (434 additional authors not shown)

**Compute**: Llama 3 405B is trained on up to 16K H100 GPUs, each running at 700W TDP with 80GB HBM3, using Meta's Grand Teton AI server platform

**Storage**: 240 PB of storage out of 7,500 servers equipped with SSDs, and supports a sustainable throughput of 2 TB/s and a peak throughput of 7 TB/s.

| Component | Category | Interruption Count | % of Interruptio |
|---|---|---|---|
| Faulty GPU | GPU | 148 | 30.1% |
| GPU HBM3 Memory | GPU | 72 | 17.2% |
| Software Bug | Dependency | 54 | 12.9% |
| Network Switch/Cable | Network | 35 | 8.4% |
| Host Maintenance | Unplanned Maintenance | 32 | 7.6% |
| GPU SRAM Memory | GPU | 19 | 4.5% |
| GPU System Processor | GPU | 17 | 4.1% |
| NIC | Host | 7 | 1.7% |
| NCCL Watchdog Timeouts | Unknown | 7 | 1.7% |
| Silent Data Corruption | GPU | 6 | 1.4% |
| GPU Thermal Interface + Sensor | GPU | 6 | 1.4% |
| SSD | Host | 3 | 0.7% |
| Power Supply | Host | 3 | 0.7% |
| Server Chassis | Host | 2 | 0.5% |
| IO Expansion Board | Host | 2 | 0.5% |
| Dependency | Dependency | 2 | 0.5% |
| CPU | Host | 2 | 0.5% |
| System Memory | Host | 2 | 0.5% |

**Table 5 Root-cause categorization of unexpected interruptions during a 54-day period of Llama 3 405B pre-training.** About 78% of unexpected interruptions were attributed to confirmed or suspected hardware issues.

| GPUs | TP | CP | PP | DP | Seq. Len. | Batch size/DP | Tokens/Batch | TFLOPs/GPU | BF16 MFU |
|---|---|---|---|---|---|---|---|---|---|
| 8,192 | 8 | 1 | 16 | 64 | 8,192 | 32 | 16M | 430 | 43% |
| 16,384 | 8 | 1 | 16 | 128 | 8,192 | 16 | 16M | 400 | 41% |
| 16,384 | 8 | 16 | 16 | 4 | 131,072 | 16 | 16M | 380 | 38% |

**Table 4 Scaling configurations and MFU for each stage of Llama 3 405B pre-training.** See text and Figure 5 for descriptions of each type of parallelism.

| | | | |
|---|---|---|---|
| Layers | 32 | 80 | 126 |
| Model Dimension | 4,096 | 8192 | 16,384 |
| FFN Dimension | 14,336 | 28,672 | 53,248 |
| Attention Heads | 32 | 64 | 128 |
| Key/Value Heads | 8 | 8 | 8 |
| Peak Learning Rate | $3 \times 10^{-4}$ | $1.5 \times 10^{-4}$ | $8 \times 10^{-5}$ |
| Activation Function | SwiGLU | | |
| Vocabulary Size | 128,000 | | |
| Positional Embeddings | RoPE ($\theta = 500,000$) | | |

# Code Example
# Q&A