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

https://www.cloudflare.com/zh-tw/learning/ai/natural-language-processing-nlp/

大型語言模型 (LLM)

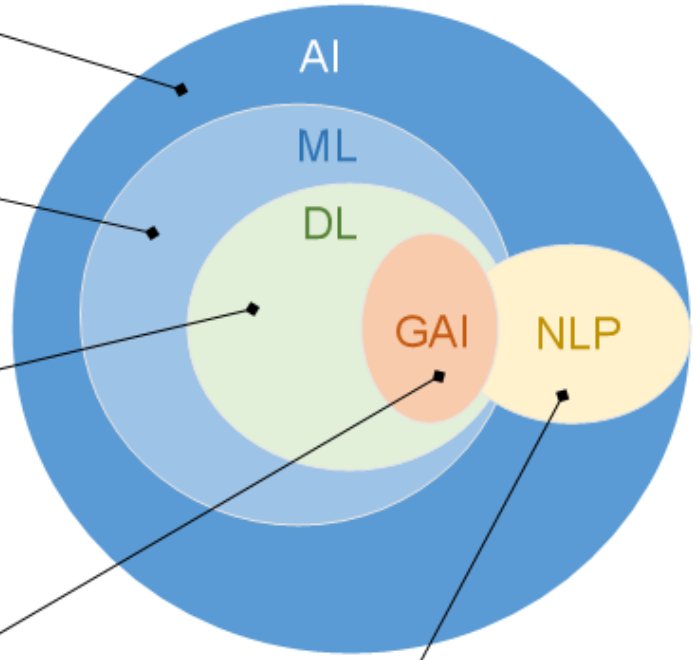
生成式人工智慧(GAI)

人工智慧(AI):利用電腦模仿人類智慧的技術

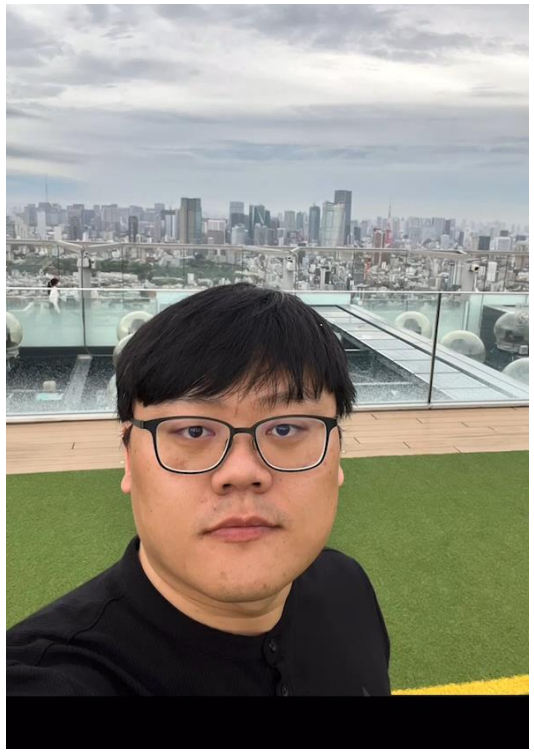
機器學習(ML):利用數據進行分析與預測的技術

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



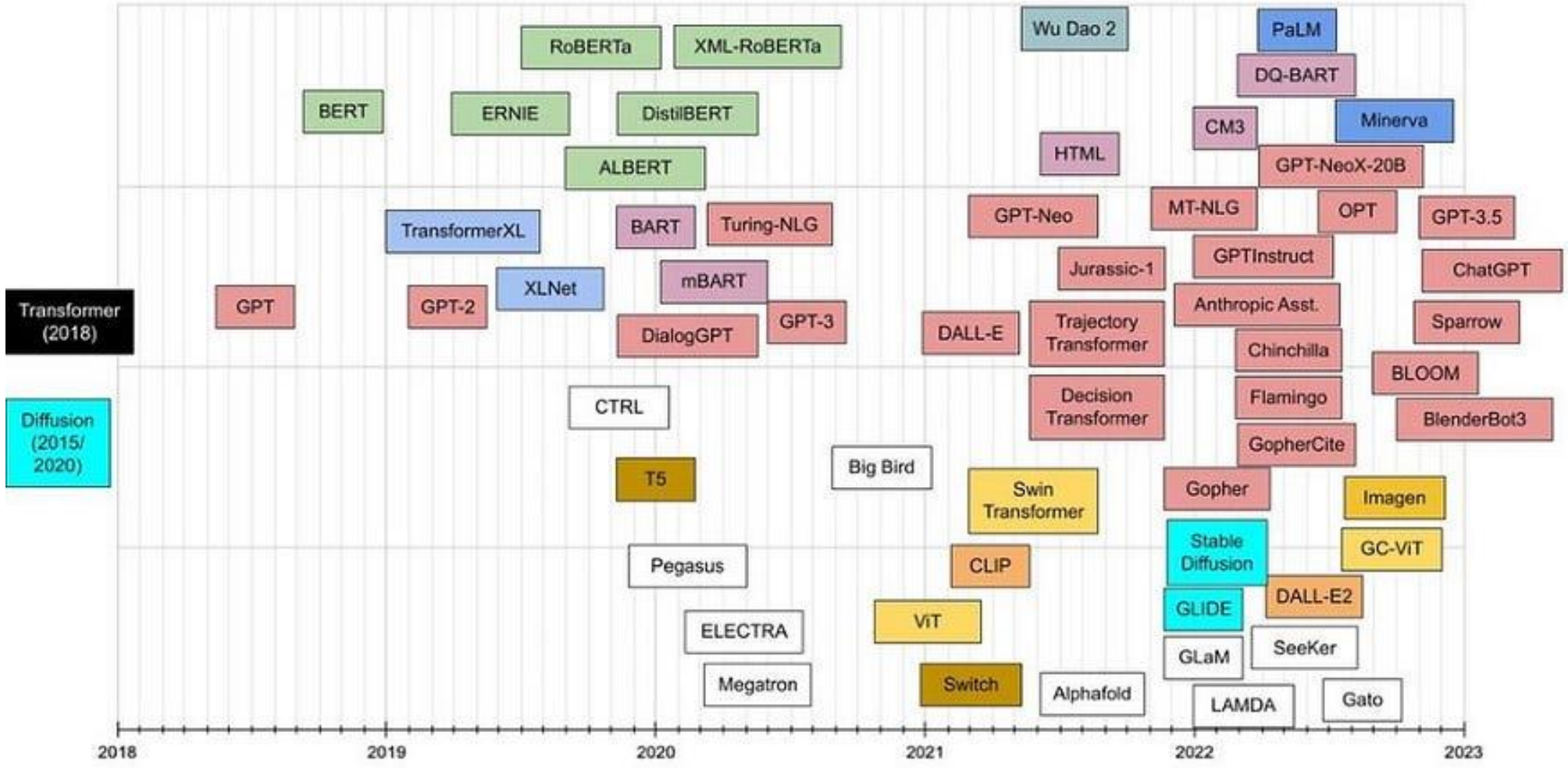
自然語言(NLP):人工智慧與語言學的分支，專注於人類語言的認知、理解、生成等



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)

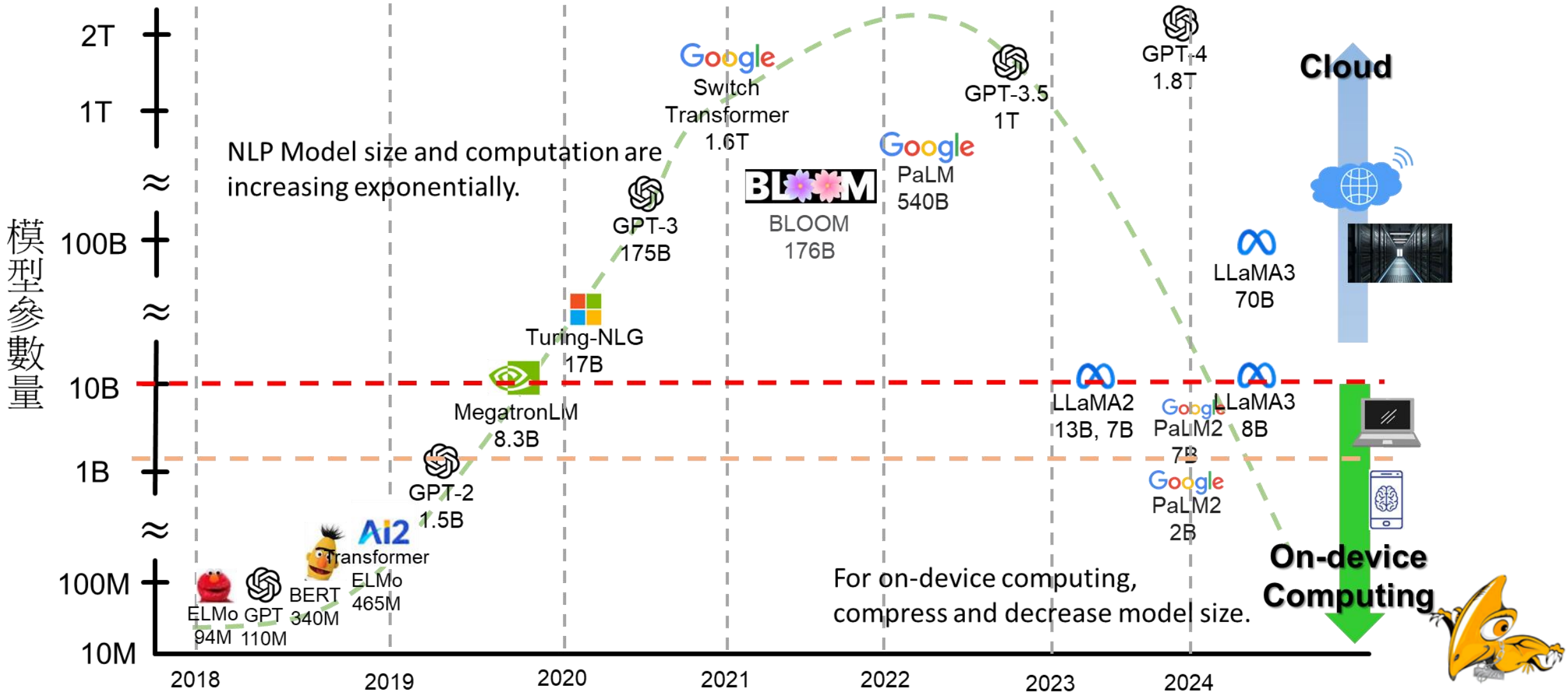
語言模型的發展歷程



https://medium.com/aigeneral/%E5%A4%A7%E5%9E%8B%E8%AA%9E%E8%A8%80%E6%A8%A1%E5%9E%8B-llm-%E7%9A%84%E7%99%BC%E5%B1%95-3da24f833020

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。



AI怎麼運作

1. Model architecture

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

Most of academia

3. Data

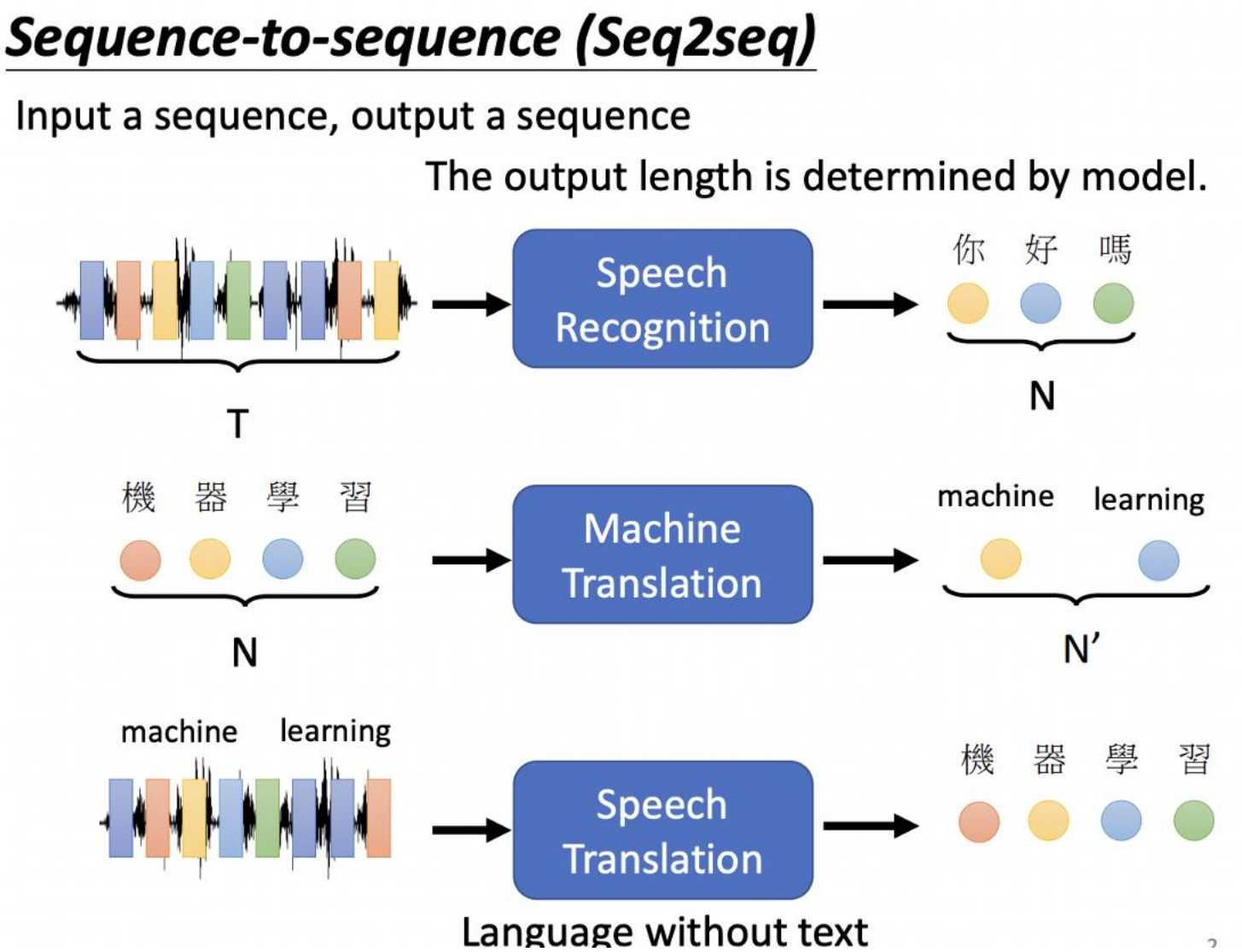
4. Evaluation

5. System:
   server architecture
   edge device
   operations,…

What matters in practice

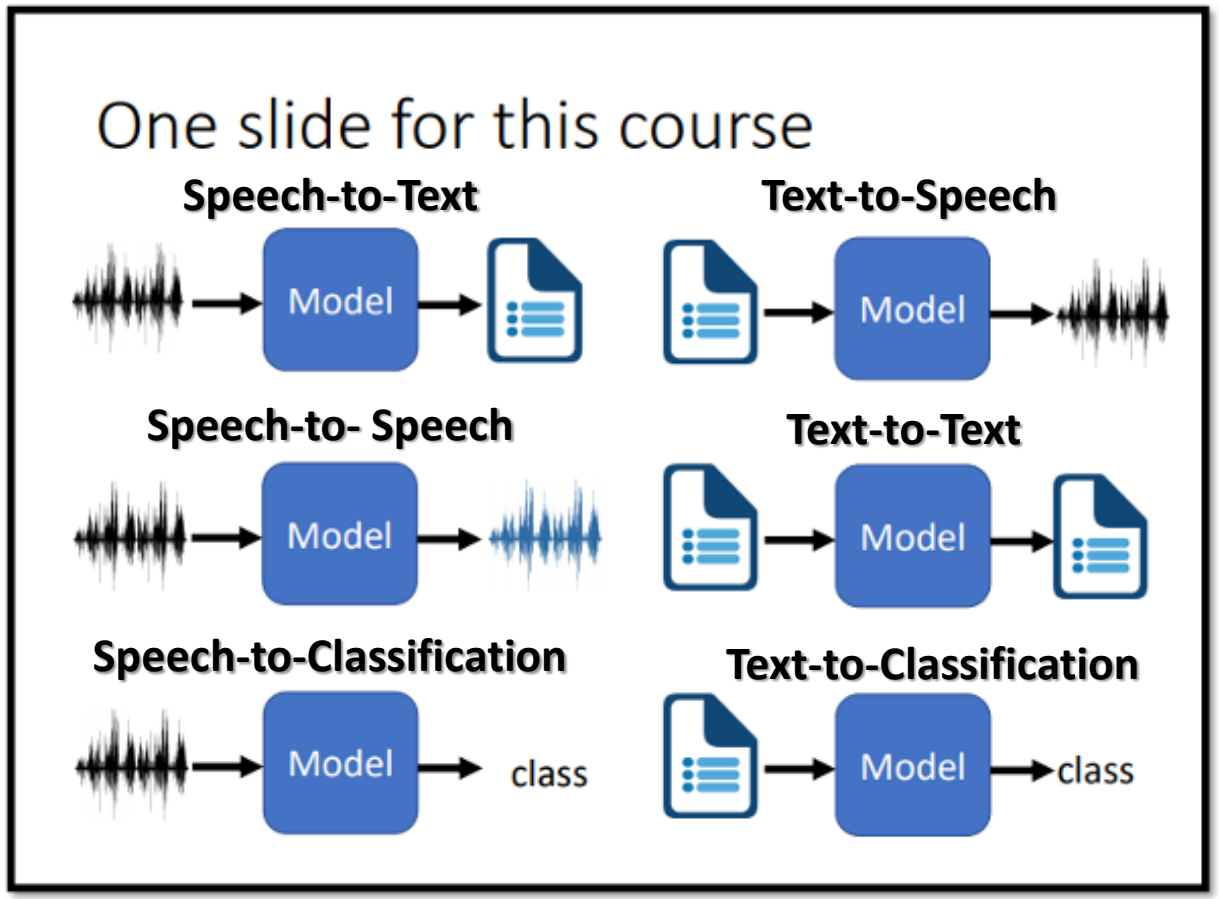
Reference: Stanford CS229 I Machine Learning I Building Large Language Models (LLMs)

Sequence-to-sequence



圖片：李宏毅老師

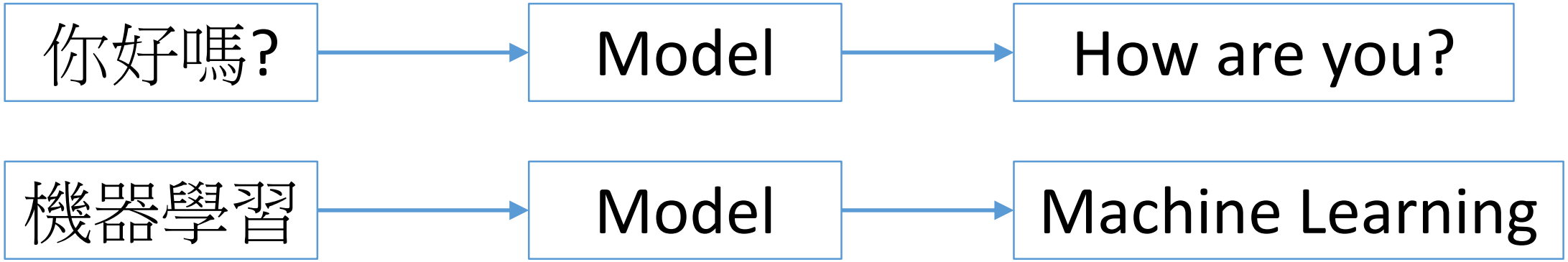
Deep Learning for Human Language Processing
深度學習與人類語言處理



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



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

- 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”、“?”

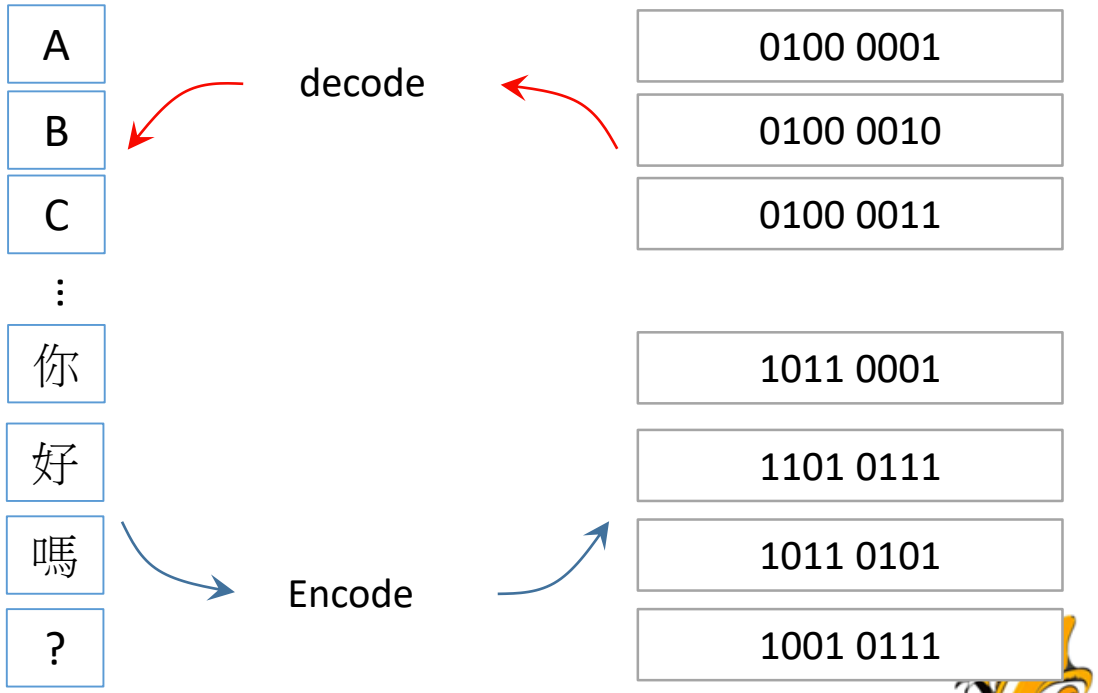
Tokenization/Tokenizer

編碼:
ASCII
Big5
Unicode
Base64

電腦看得懂文字(中文/英文/日文/...)?

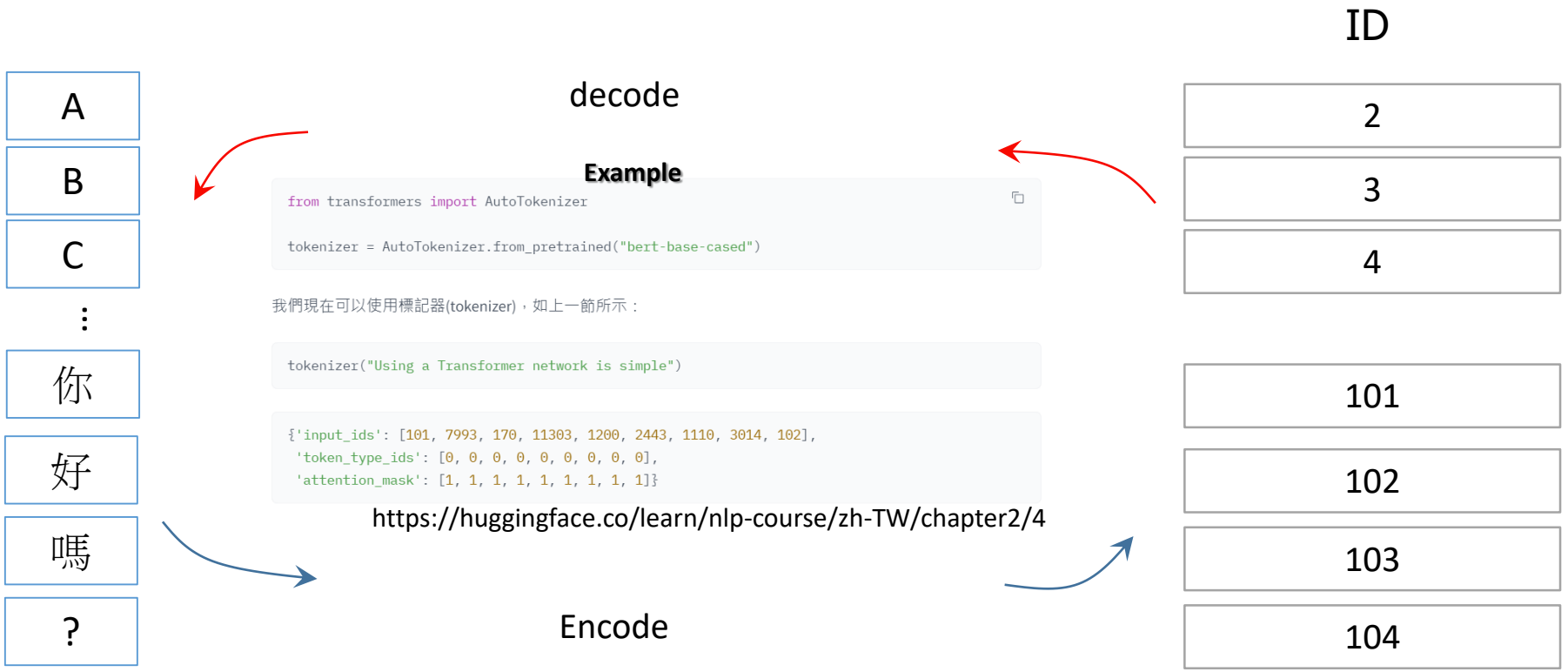
編碼: **ASCII**

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



Note:中文編碼是我亂打的

Tokenization/Tokenizer



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

Hi → [30, 7]

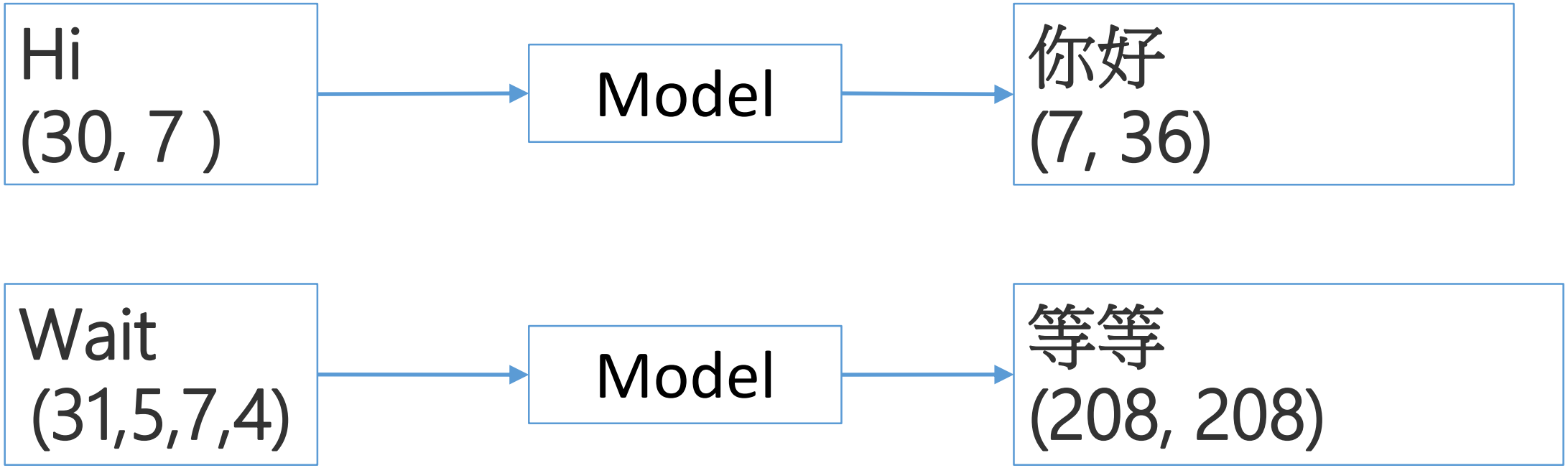
Wait → [31, 5, 7, 4]

中文有自己的字典

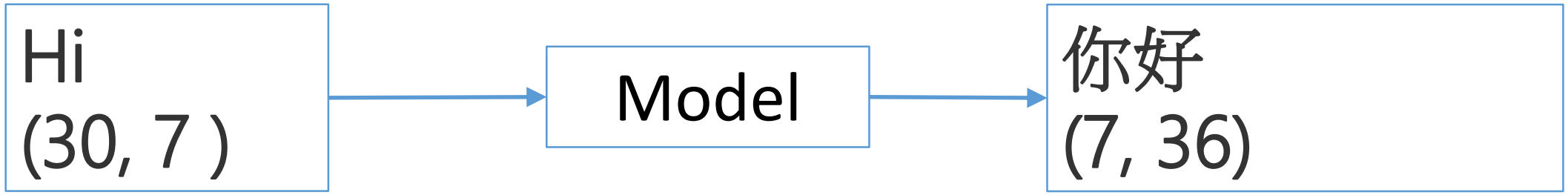
你好 → [7, 36]

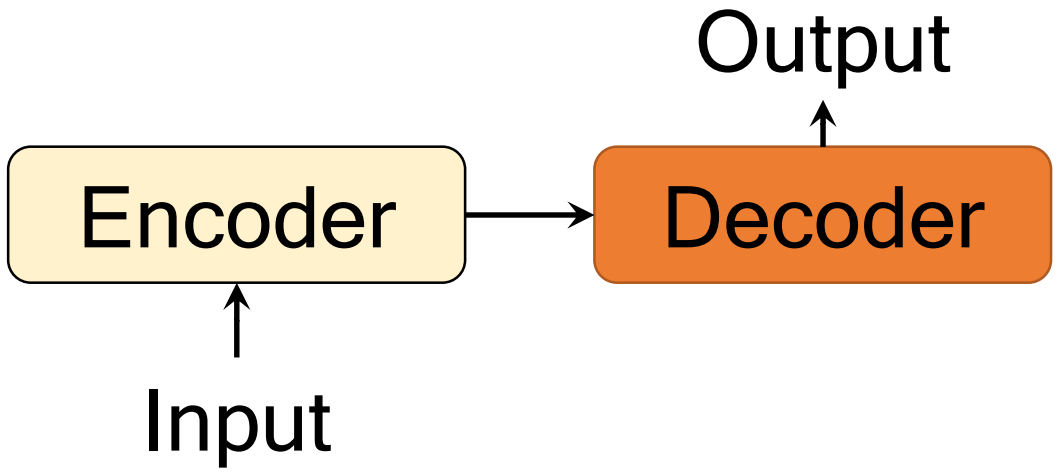
等等 → [208, 208]

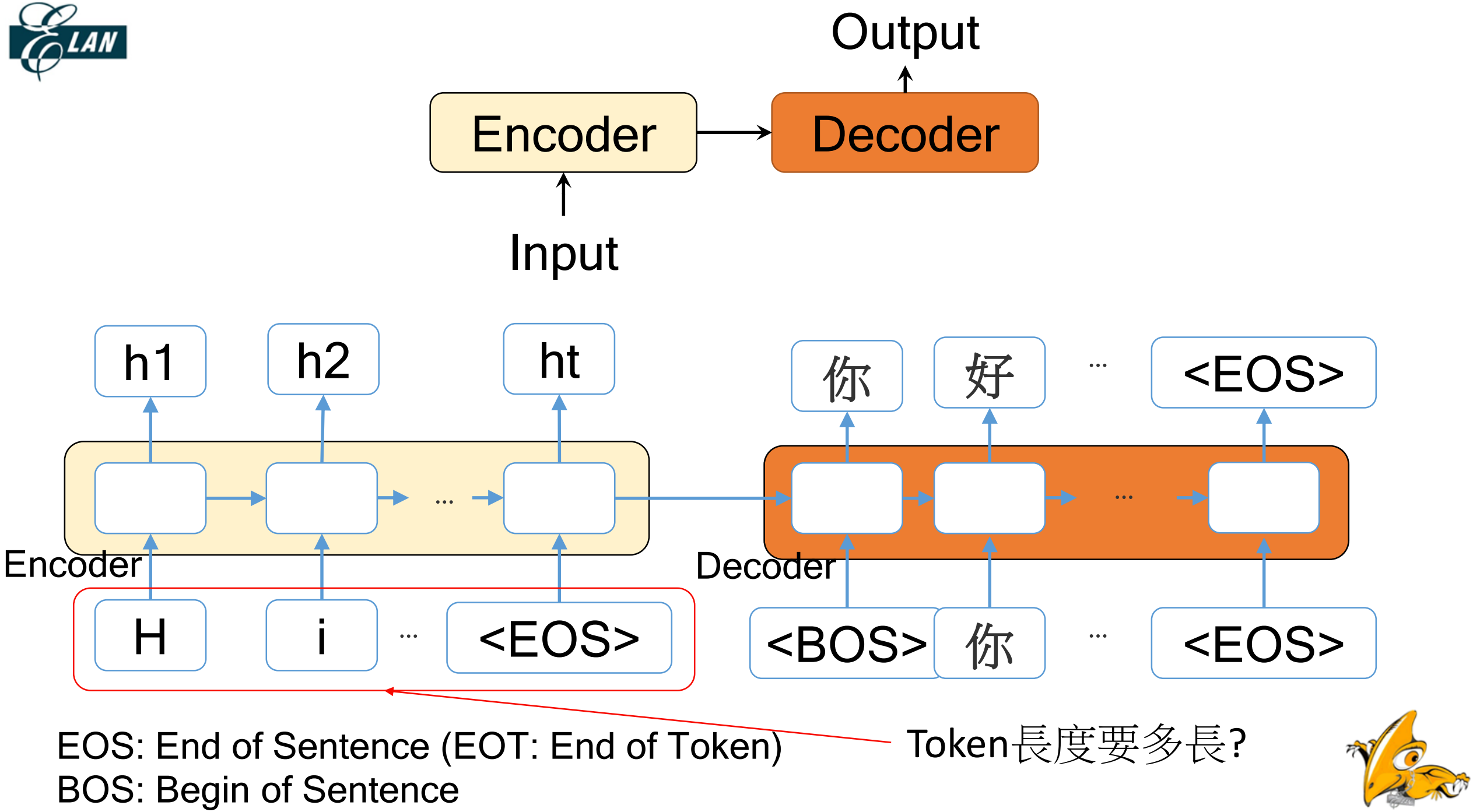
Sequence-to-sequence
Text-to-Text: 中英文翻譯

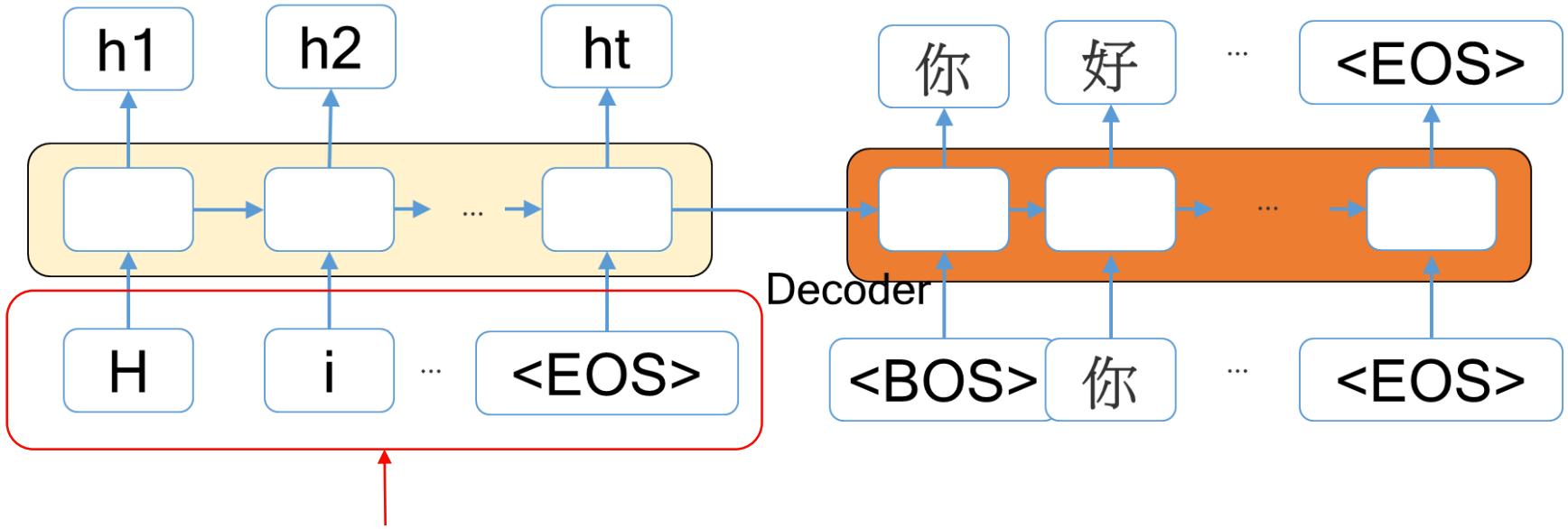


Sequence-to-sequence






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?

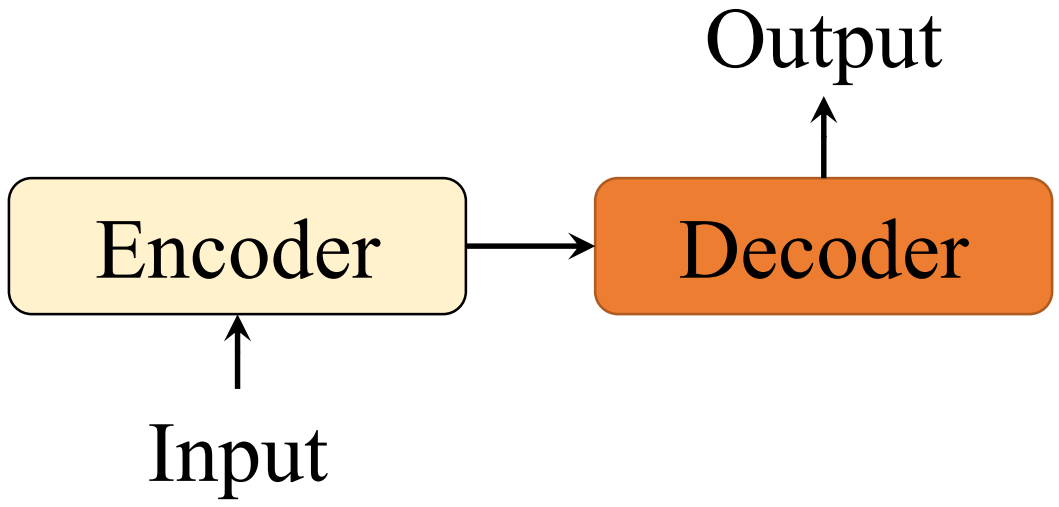
The previous example runs until you hit the model's token limit. With each question asked, and answer received, the `messages` list grows in size. The token limit fo `gpt-35-turbo` is 4,096 tokens. The token limits for `gpt-4` and `gpt-4-32k` are 8,192 and 32,768, respectively. These limits include th 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

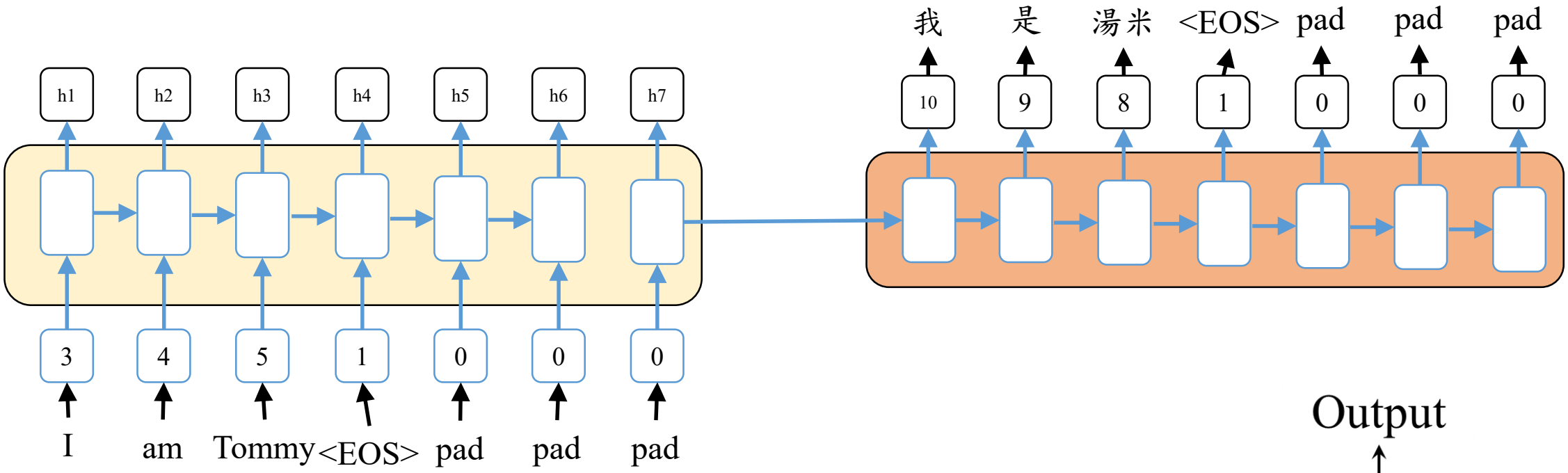


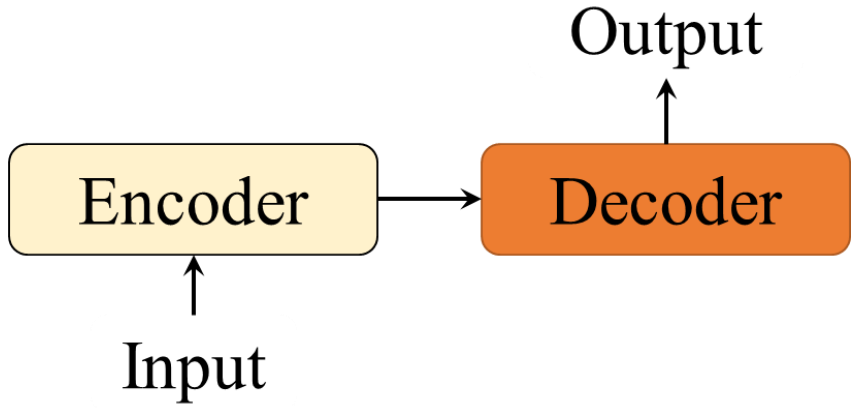
<EOS> : 1
Pad : 0
<BOS> : 2

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]



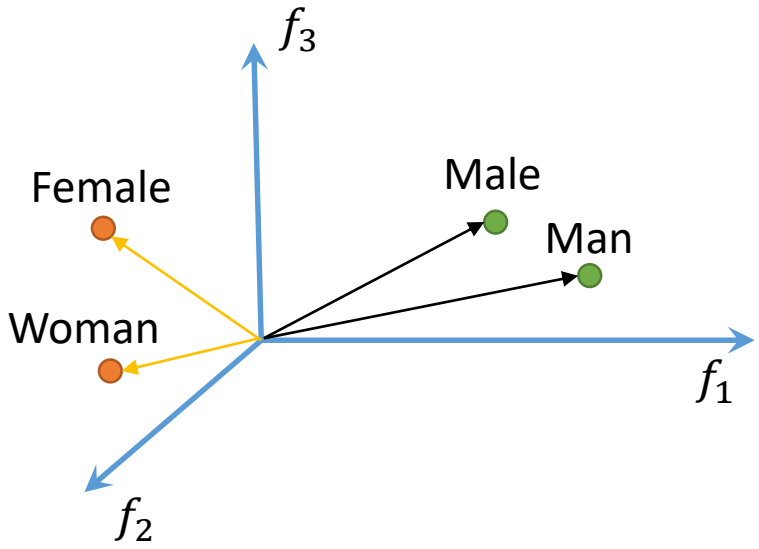


3,4,5,6→數字本身沒有意義

Tokenization: Word Embedding

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

- 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

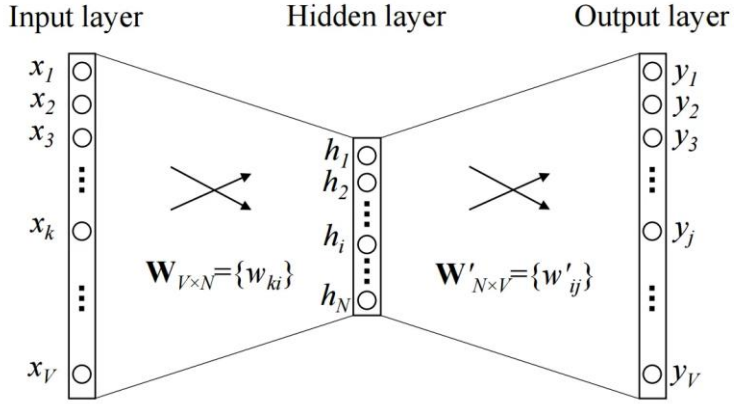
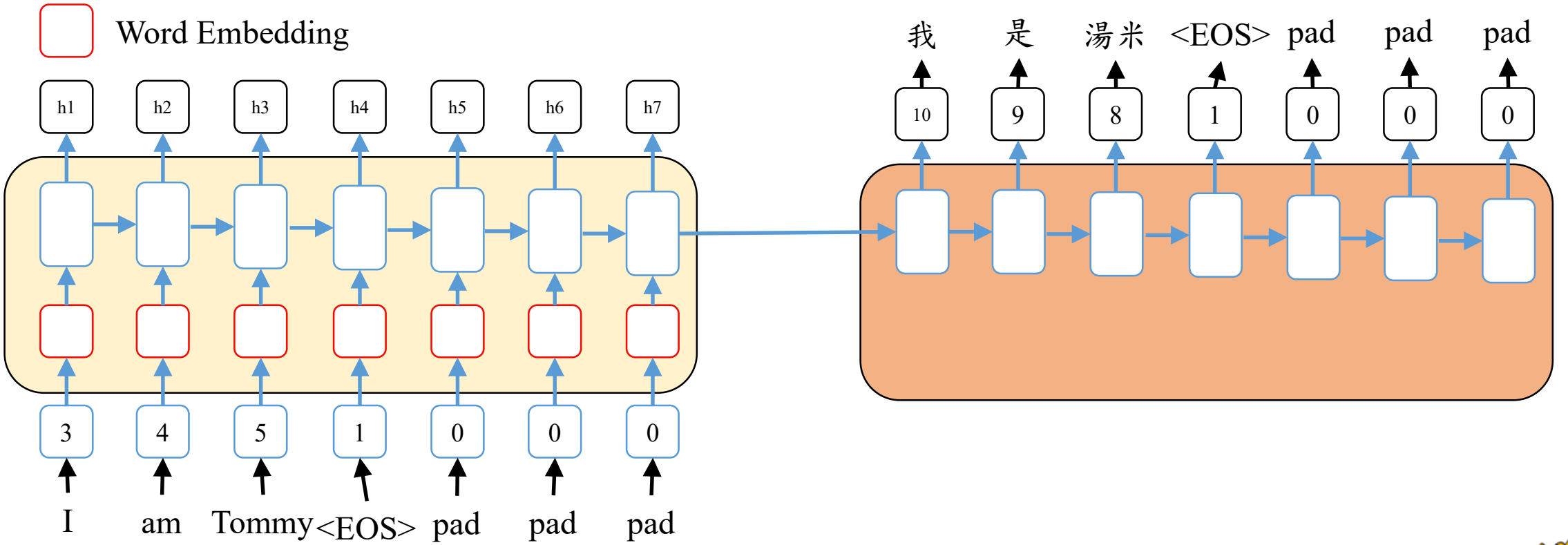


Figure 1: A simple CBOW model with only one word in the context

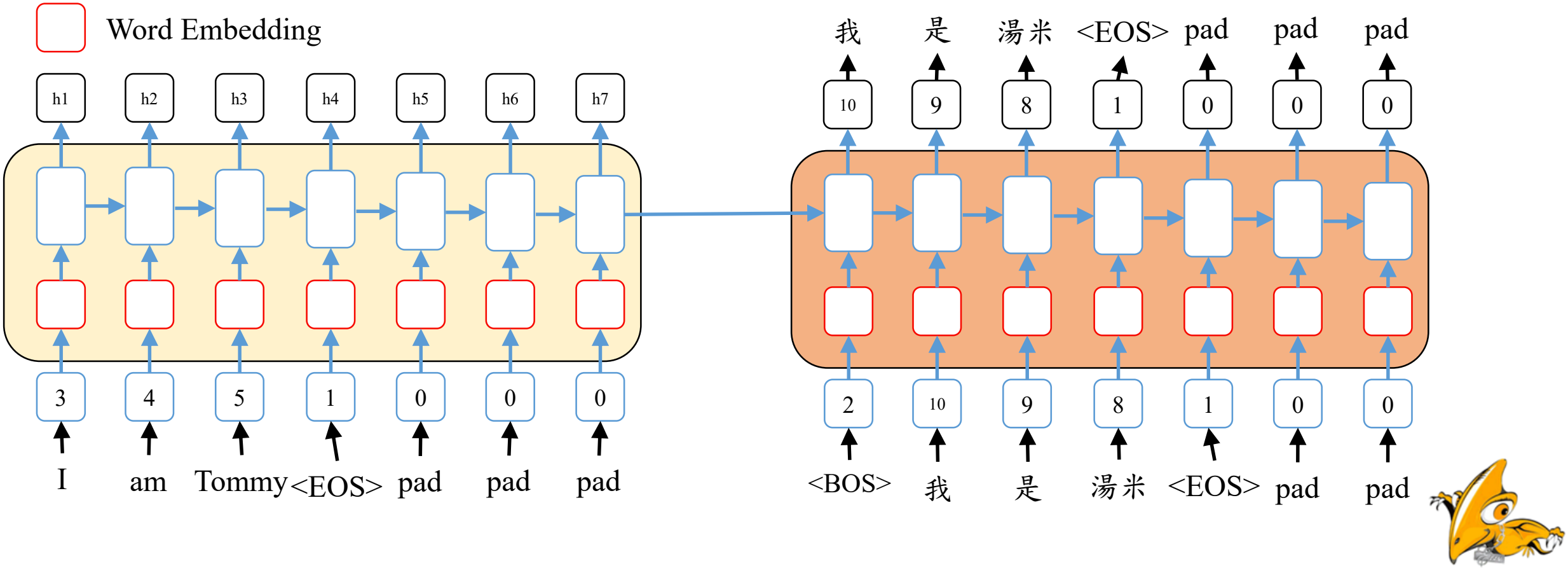
Tokenization: Word Embedding

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]



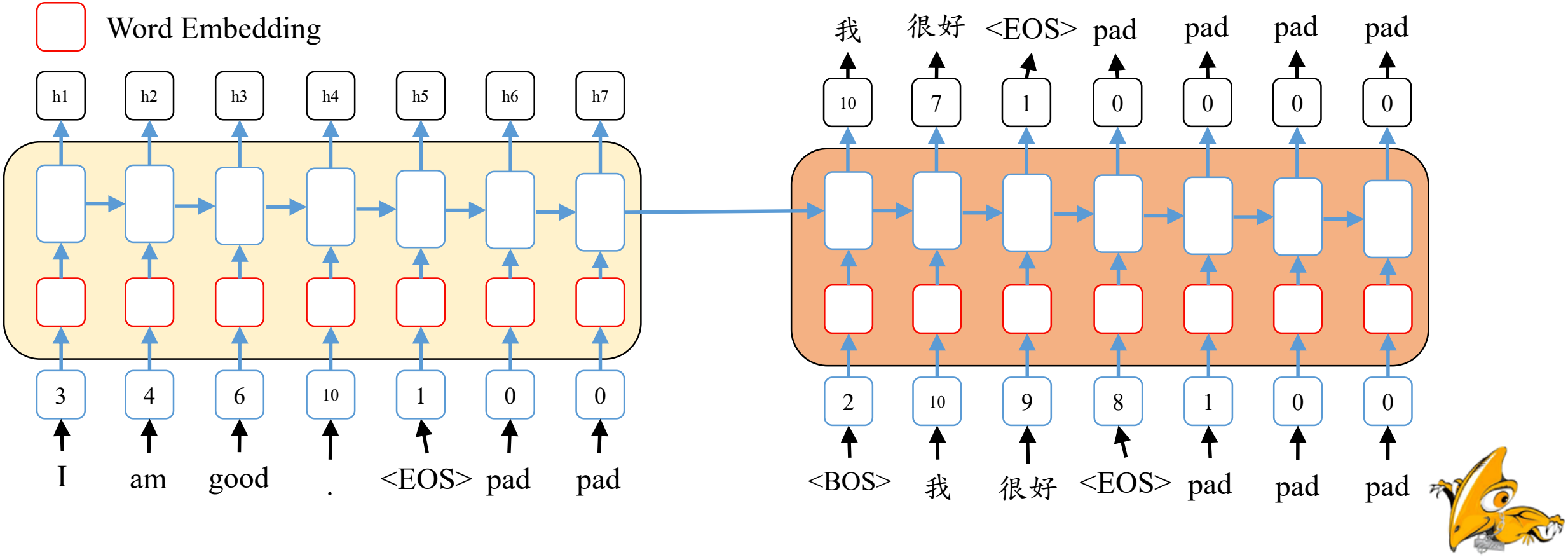
範例-I am Tommy

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]



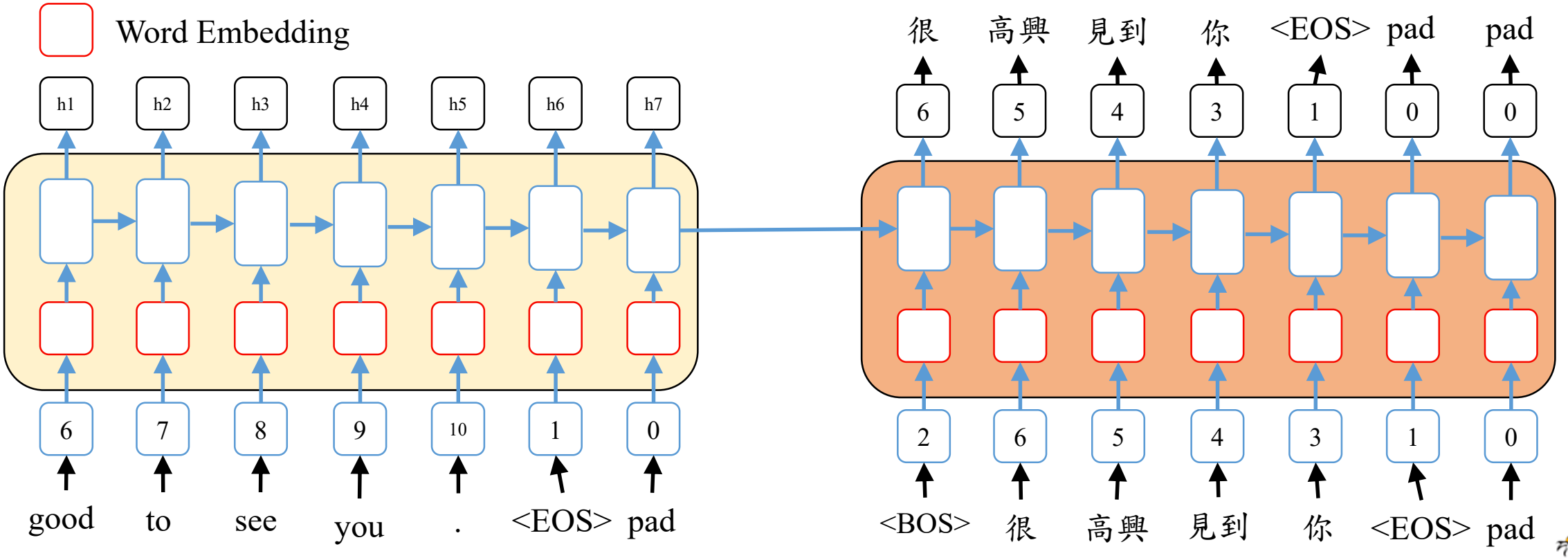
範例-I am good.

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]



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
正常	Book	Book	101
	is	is	2505
	good	good	500
打錯	booook	Unknown	?
	tarsformer	Unknown	?

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

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

merge

ug 50+5+5=60次

un 12+4=16次

pu 5+12=17次

hu 10+5=15次

字典/Corpus

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

ug

只是舉例用，
第一個iteration應
該是算不到這步

Tokenizer: Byte Pair Encoding

字典/Corpus

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

ug

un

頻率	
10	h ug
5	p ug
12	p u n
4	b u n
5	h ug s

merge

un 12+4=16次

hug 10+5=15次

Tokenizer: Byte Pair Encoding

頻率	
10	h ug
5	p ug
12	p un
4	b un
5	h ug s

merge

hug 10+5=15次

字典/Corpus

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

ug

un

hug

Tokenizer: Byte Pair Encoding

頻率		merge	
10	hug		
5	p ug	pug	5次
12	p un	pug	12次
4	b un	bun	4次
5	hug s	hugs	5次

字典/Corpus

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

ug

un

hug

pug

pun

hugs

bun

Tokenizer: Byte Pair Encoding

頻率	
10	hug
5	p ug
12	p un
4	b un
5	hug s

字典/Corpus

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

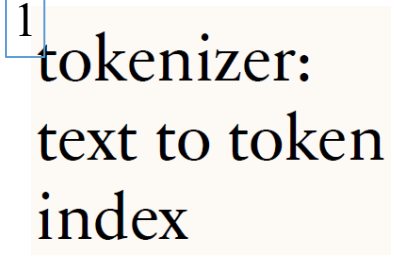
Idea: tokens as common subsequences

Byte Pair Encoding (BPE). Train steps:

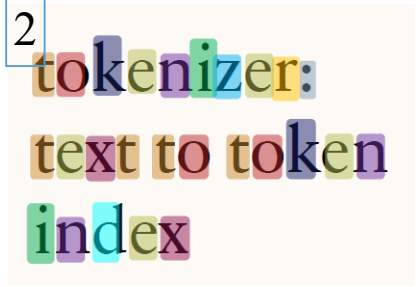
1. Take large corpus of text
2. Start with one token per character
3. Merge common pairs of tokens into a token
4. Repeat until desired vocab size or all merged

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

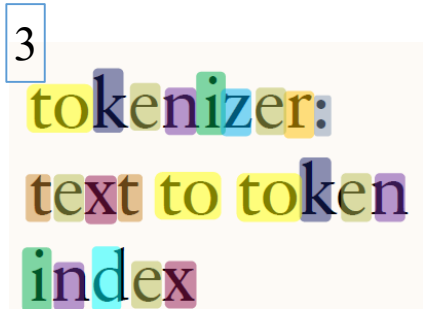
1



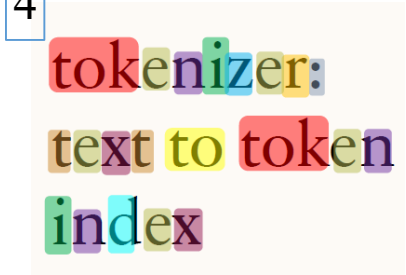
2

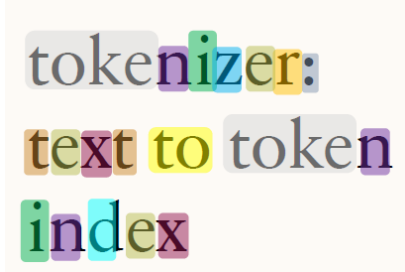


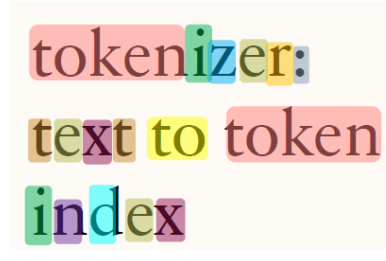
3

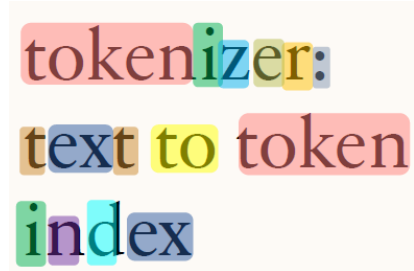


4







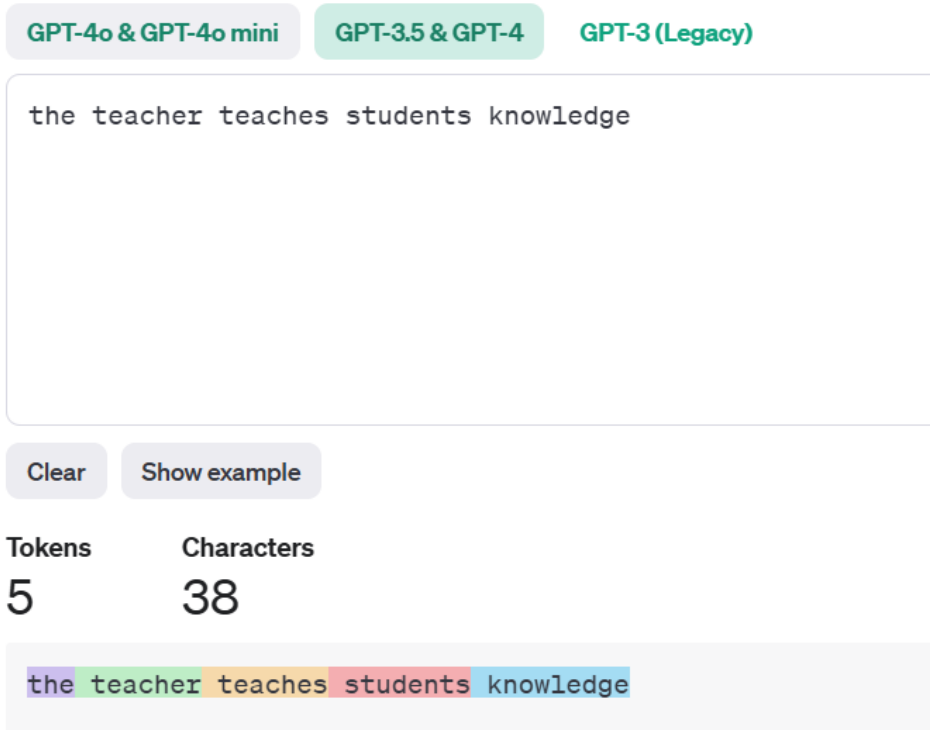


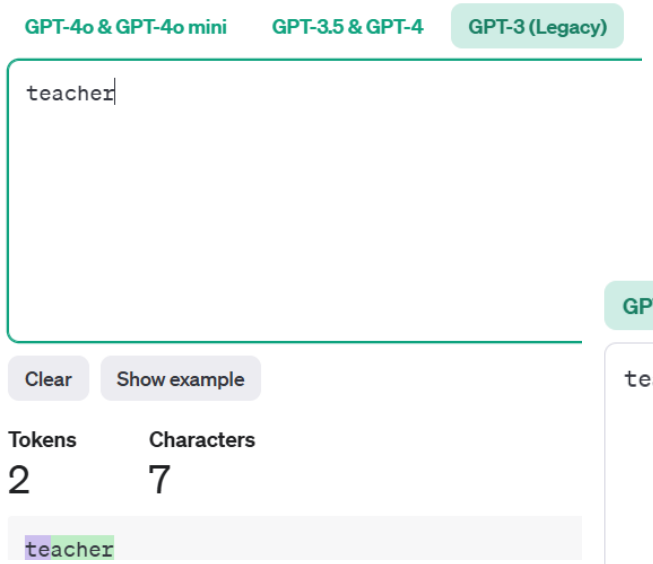
Real Tokenizer in GPT

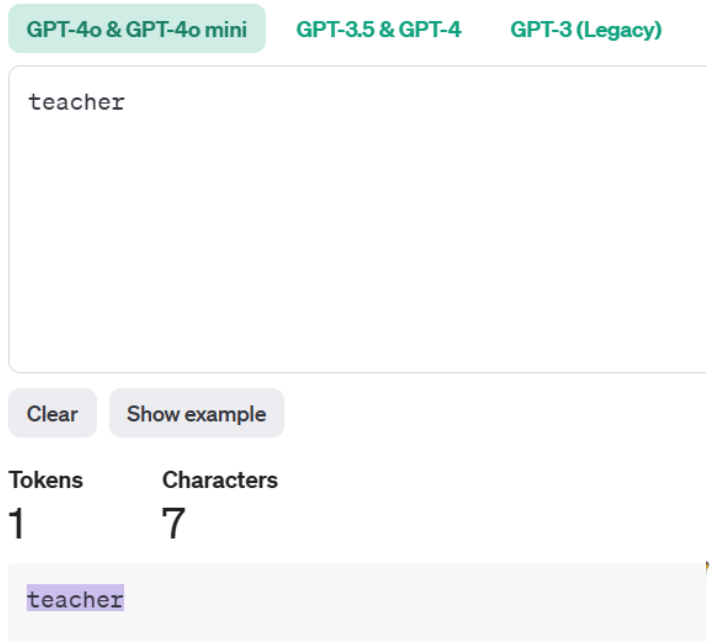
Reference: Stanford CS229 I Machine Learning I Building Large Language Models (LLMs)

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

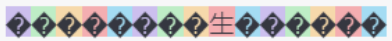
OpenAI Platform

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

老師教學生知識

Clear | Show example

Tokens: 15 | Characters: 7



Text | Token IDs

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

老師教學生知識

Clear | Show example

Tokens: 15 | Characters: 7

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

Text | Token IDs

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

老師教學生知識

Clear | Show example

Tokens: 5 | Characters: 7

老師教學生知識

Text | Token IDs

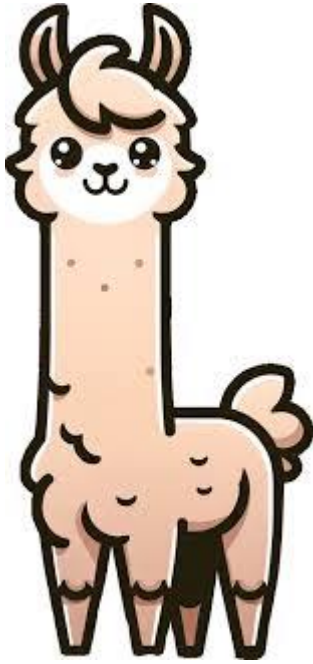
Tokenizing Chinese by Unicode encoding.

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 Interruptic
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	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Code Example Q&A