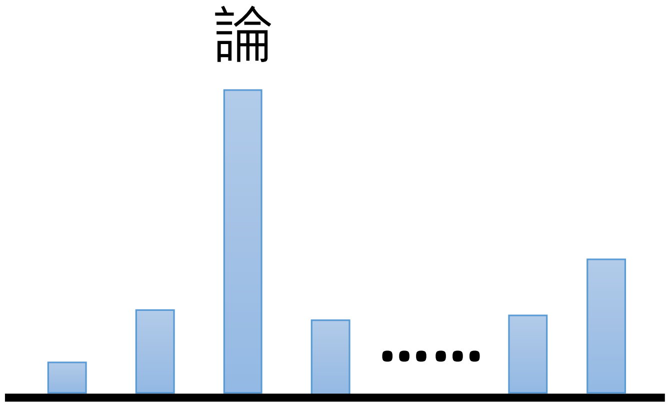
今日的語言模型
是如何做文字接龍的

Transformer 簡介

前言

這門課是生成式 AI 導

語言模型



找出參數

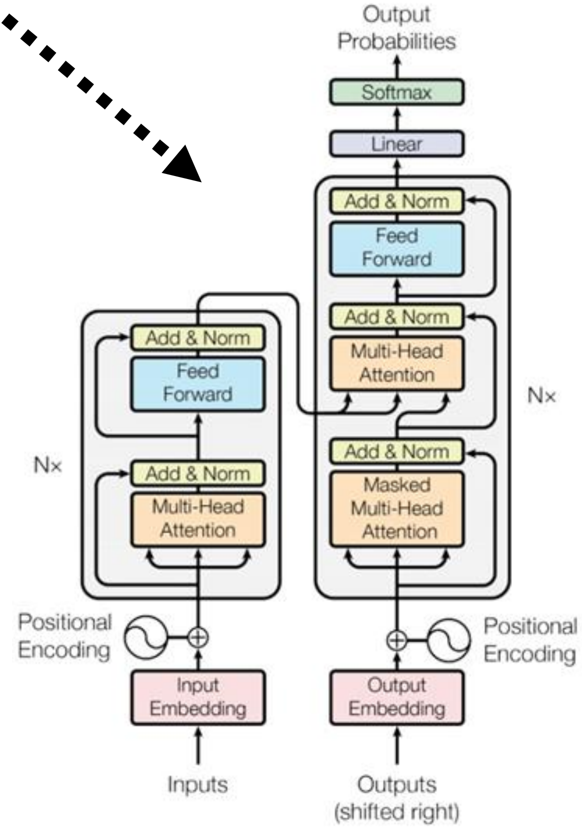
輸入：人工智　　輸出：慧

輸入：不要忘了今天來開　　輸出：會

輸入：床前明月　　輸出：光

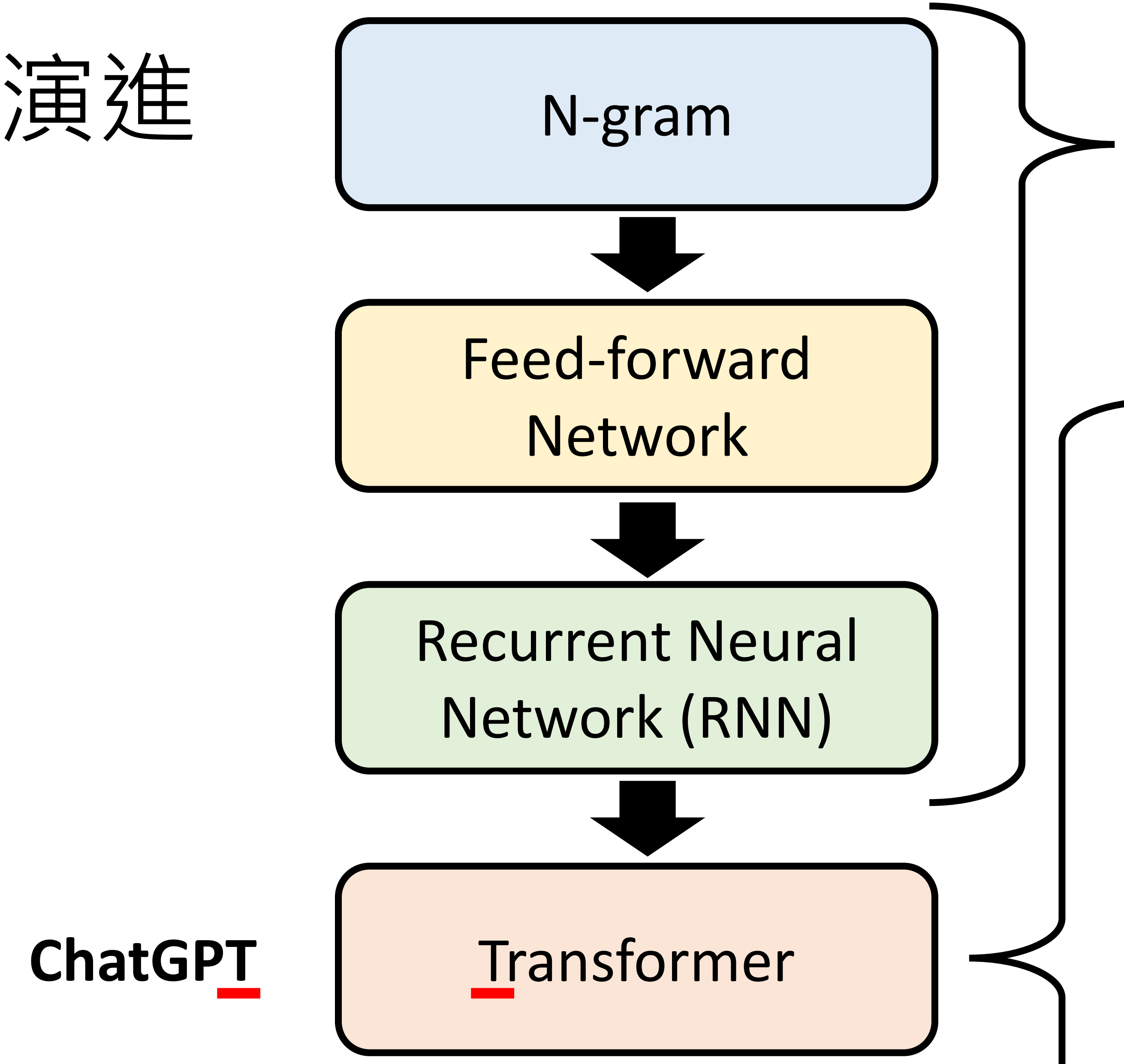
⋮　　⋮

訓練資料



Transformer

模型演進



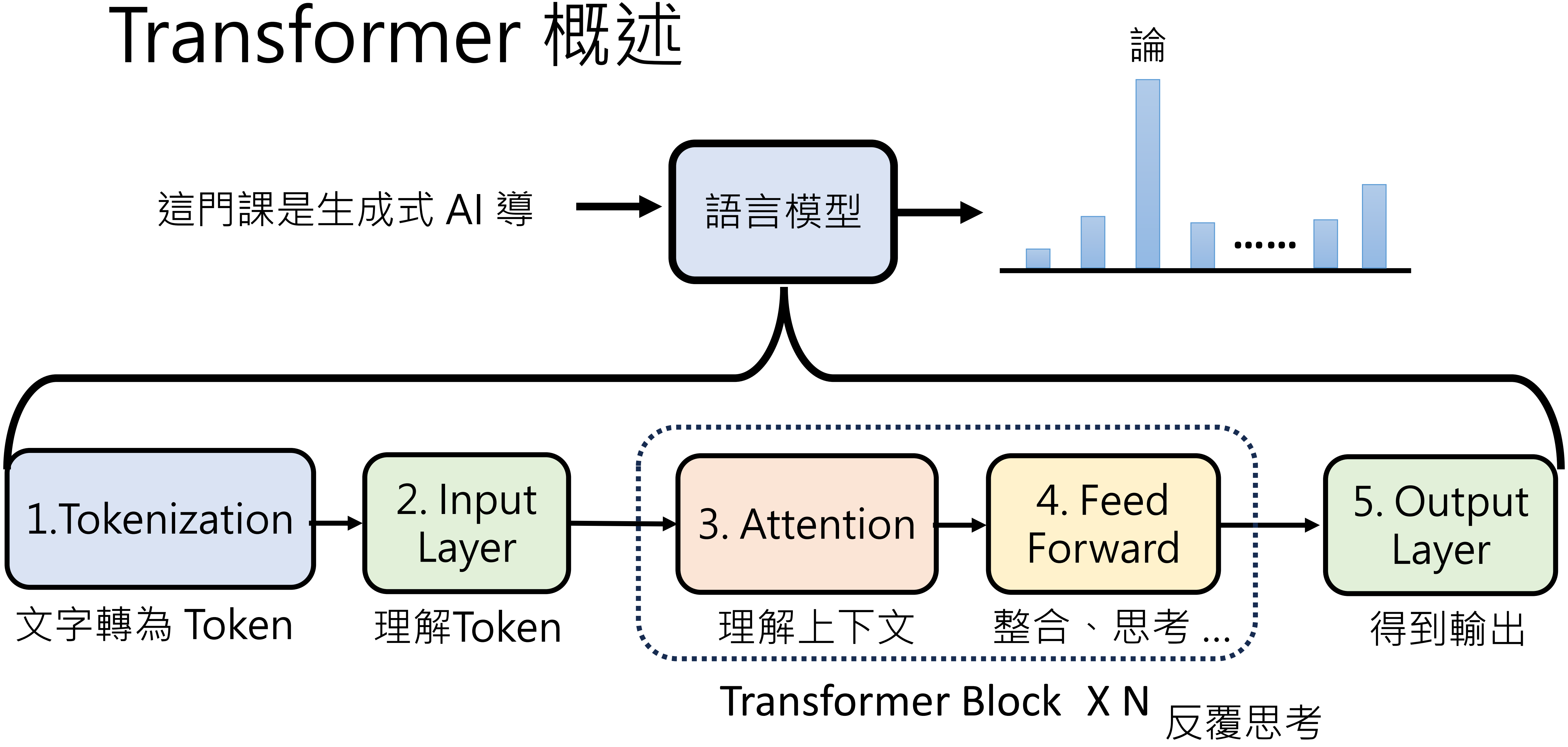
https://youtu.be/dymfkWtVUdo?si=Ng29H_YxaoeiX_4y

https://youtu.be/n9TlOhRjYoc?si=yaadpbm8w1UgbKkU

https://youtu.be/N6aRv06iv2g?si=FuemBCZt8ChwHOvu

(本課程會對 Transformer 的說明進行大量簡化，詳細內容請見過去課程)

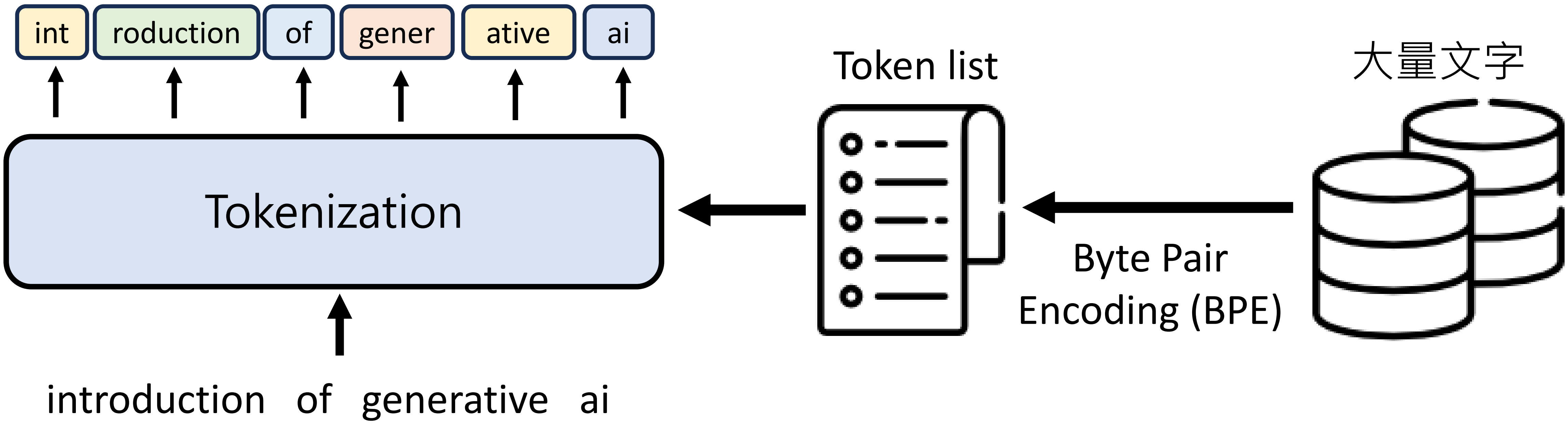
Transformer 概述



1. 把文字變成 Token

- 語言模型是以 Token 作為單位來對文字進行處理

這頁投影片沒有需要訓練的參數



int roduction of gener ative ai

Tokenization

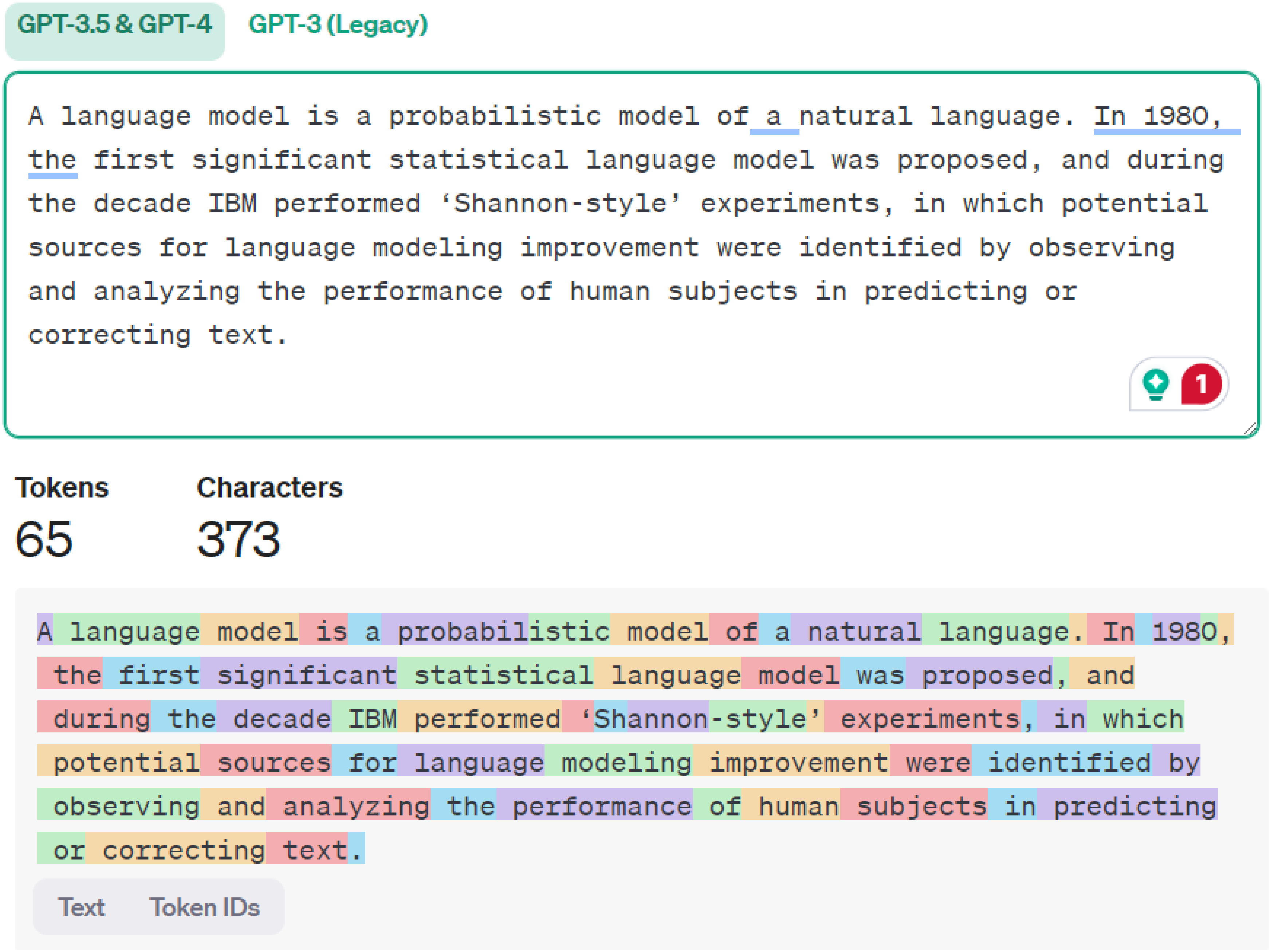
introduction of generative ai

Token list

大量文字

Byte Pair Encoding (BPE)

https://huggingface.co/learn/nlp-course/chapter6/5



A language model is a probabilistic model of a natural language. In 1980, the first significant statistical language model was proposed, and during the decade IBM performed ‘Shannon-style’ experiments, in which potential sources for language modeling improvement were identified by observing and analyzing the performance of human subjects in predicting or correcting text.

Tokens
65

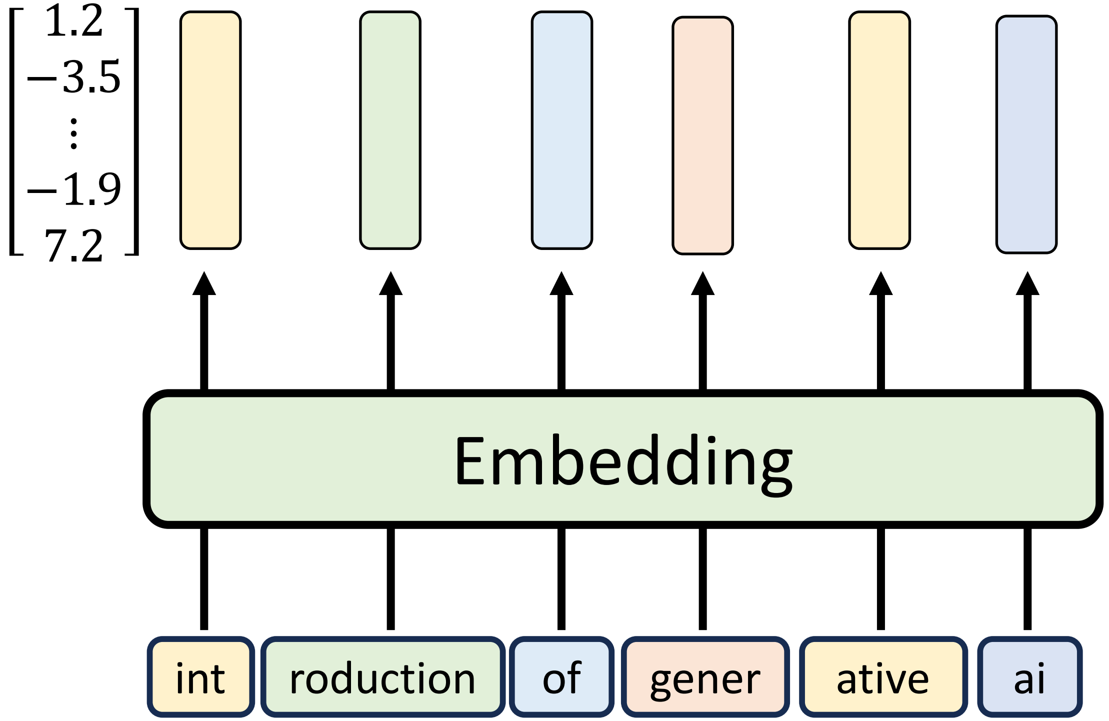
Characters
373

A language model is a probabilistic model of a natural language. In 1980, the first significant statistical language model was proposed, and during the decade IBM performed ‘Shannon-style’ experiments, in which potential sources for language modeling improvement were identified by observing and analyzing the performance of human subjects in predicting or correcting text.

https://platform.openai.com/tokenizer

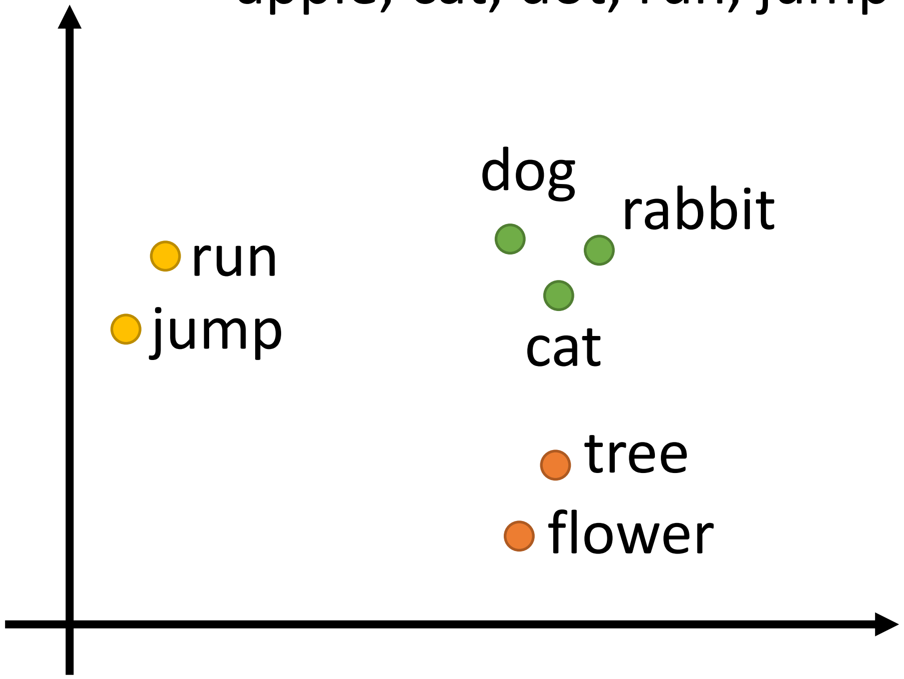
2. 理解每個 Token — 語意

向量 (Vector)



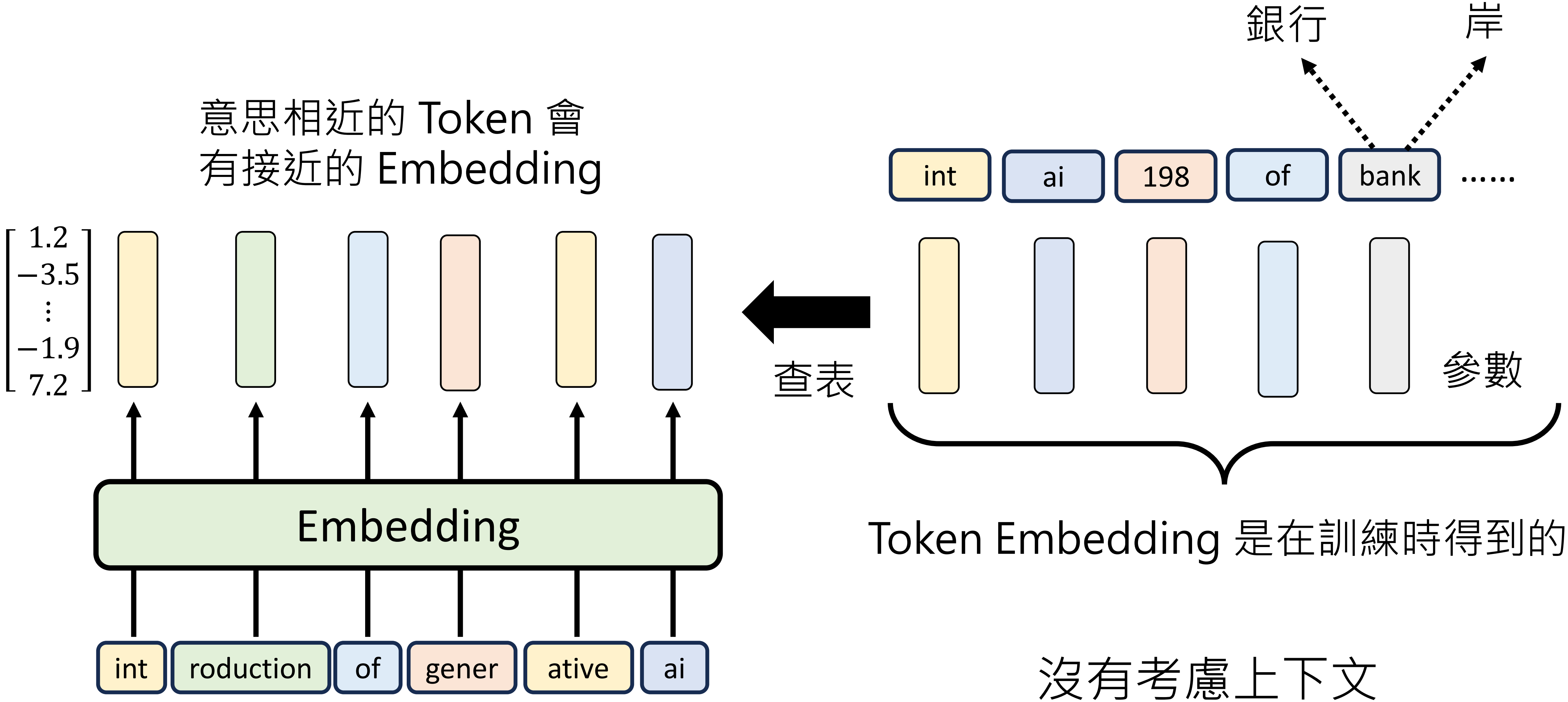
原本每一個 Token 都是獨立的符號

apple, cat, dot, run, jump ……



意思相近的 Token 會有接近的 Embedding

2. 理解每個 Token — 語意



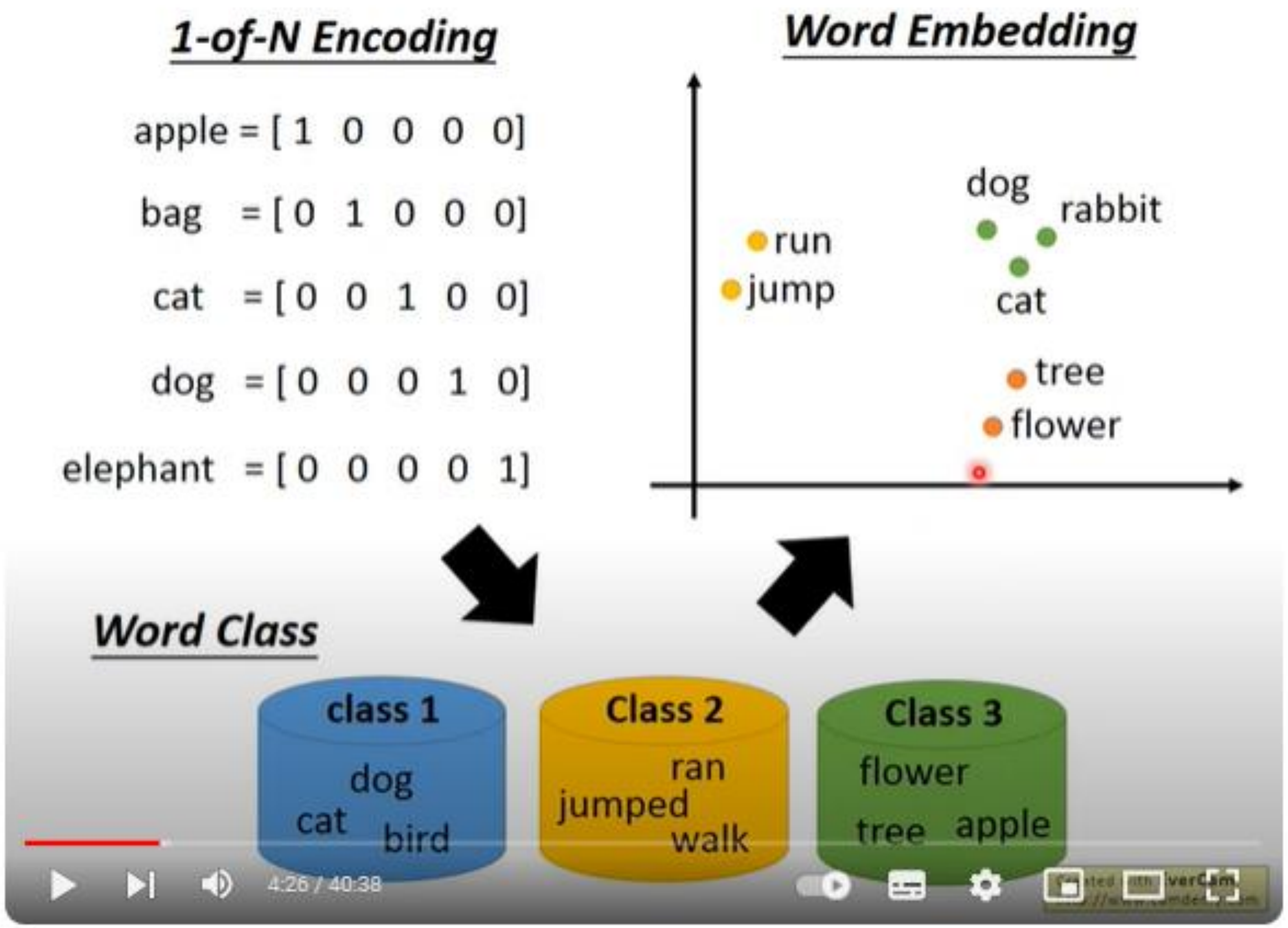
意思相近的 Token 會
有接近的 Embedding

Token Embedding 是在訓練時得到的

沒有考慮上下文

2. 理解每個 Token — 語意

- Word Embedding



1-of-N Encoding

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

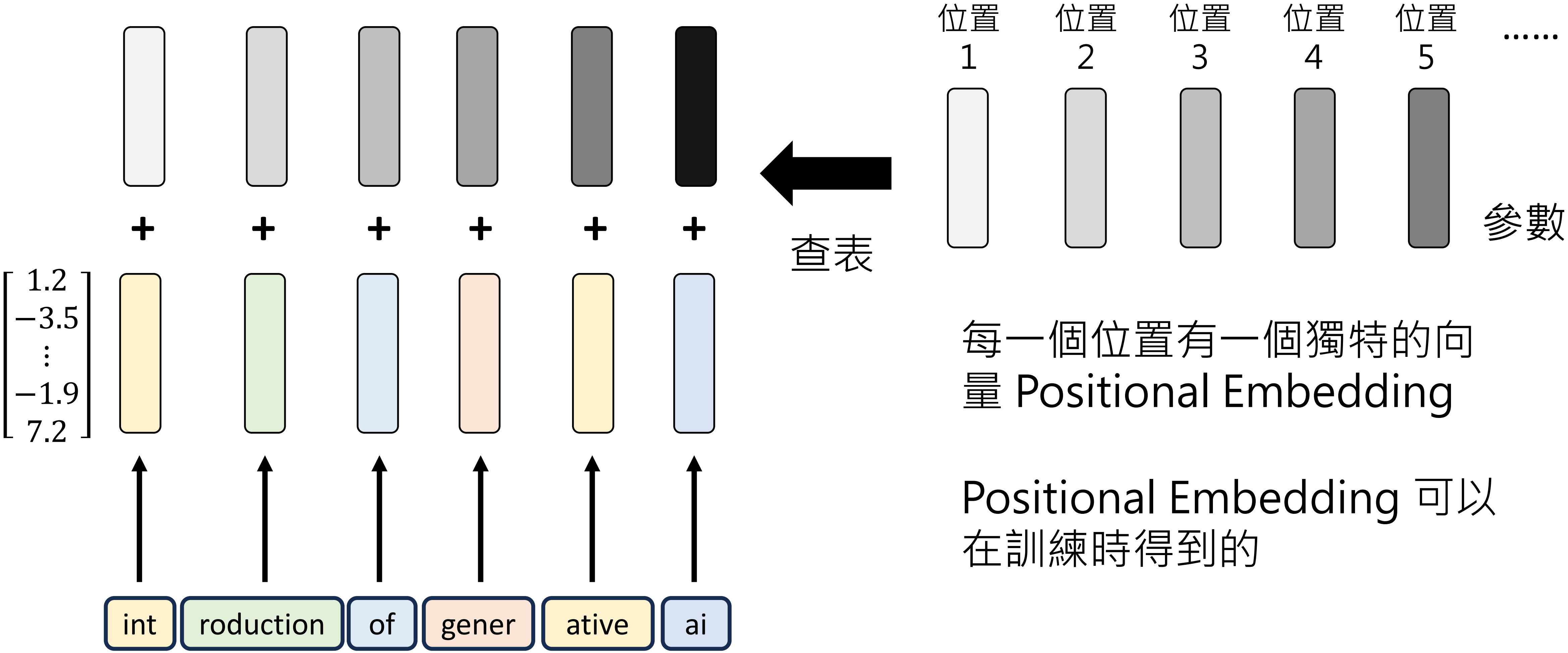
dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

ML Lecture 14: Unsupervised Learning - Word Embedding

https://youtu.be/X7PH3NuYW0Q?si=XRCVhgJAfY5a8nkf

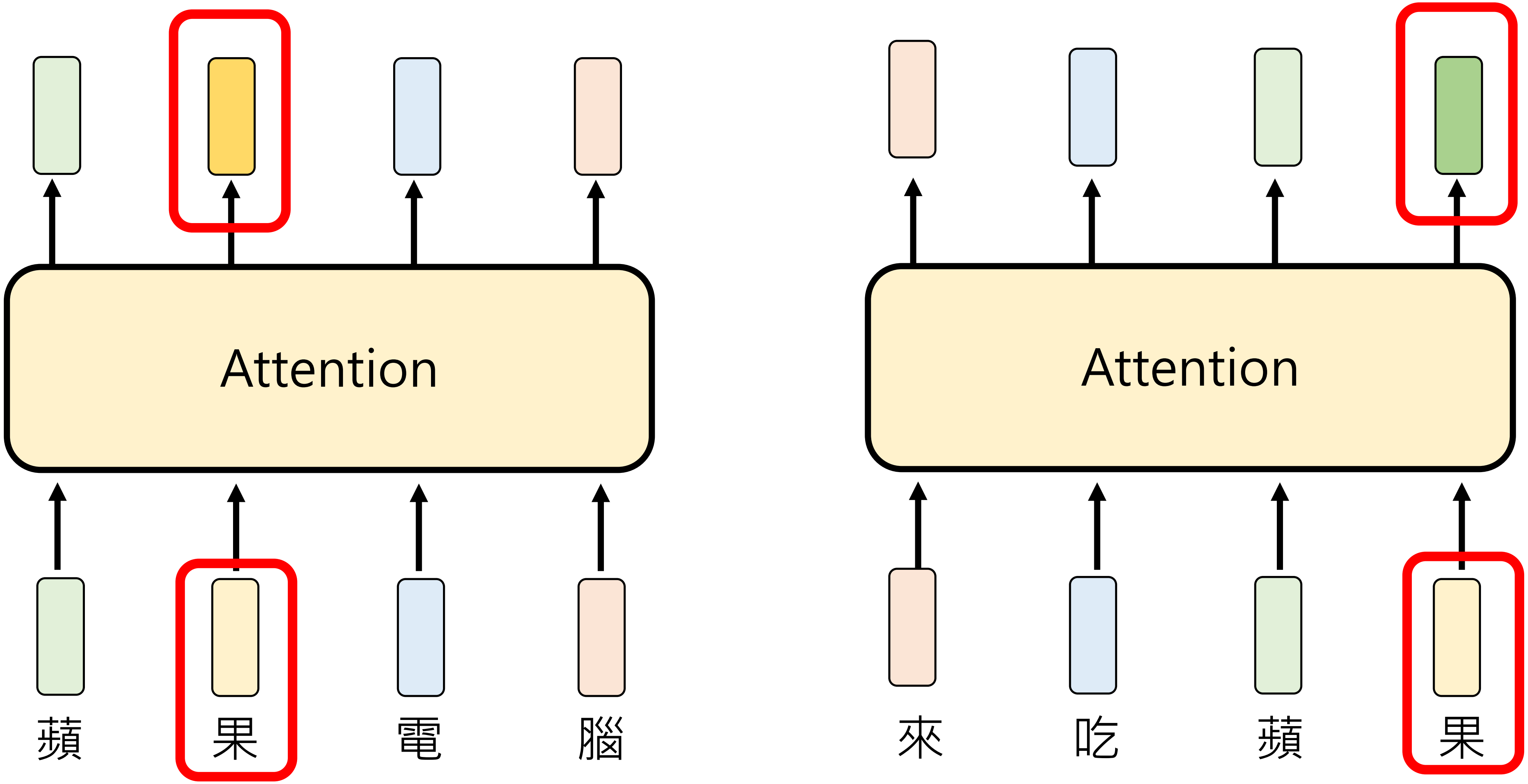
2. 理解每個 Token — 位置



每一個位置有一個獨特的向量 Positional Embedding

Positional Embedding 可以在訓練時得到的

3. Attention: 考慮上下文



3. Attention: 考慮上下文

Attention Is All You Need

https://arxiv.org/abs/1706.03762

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

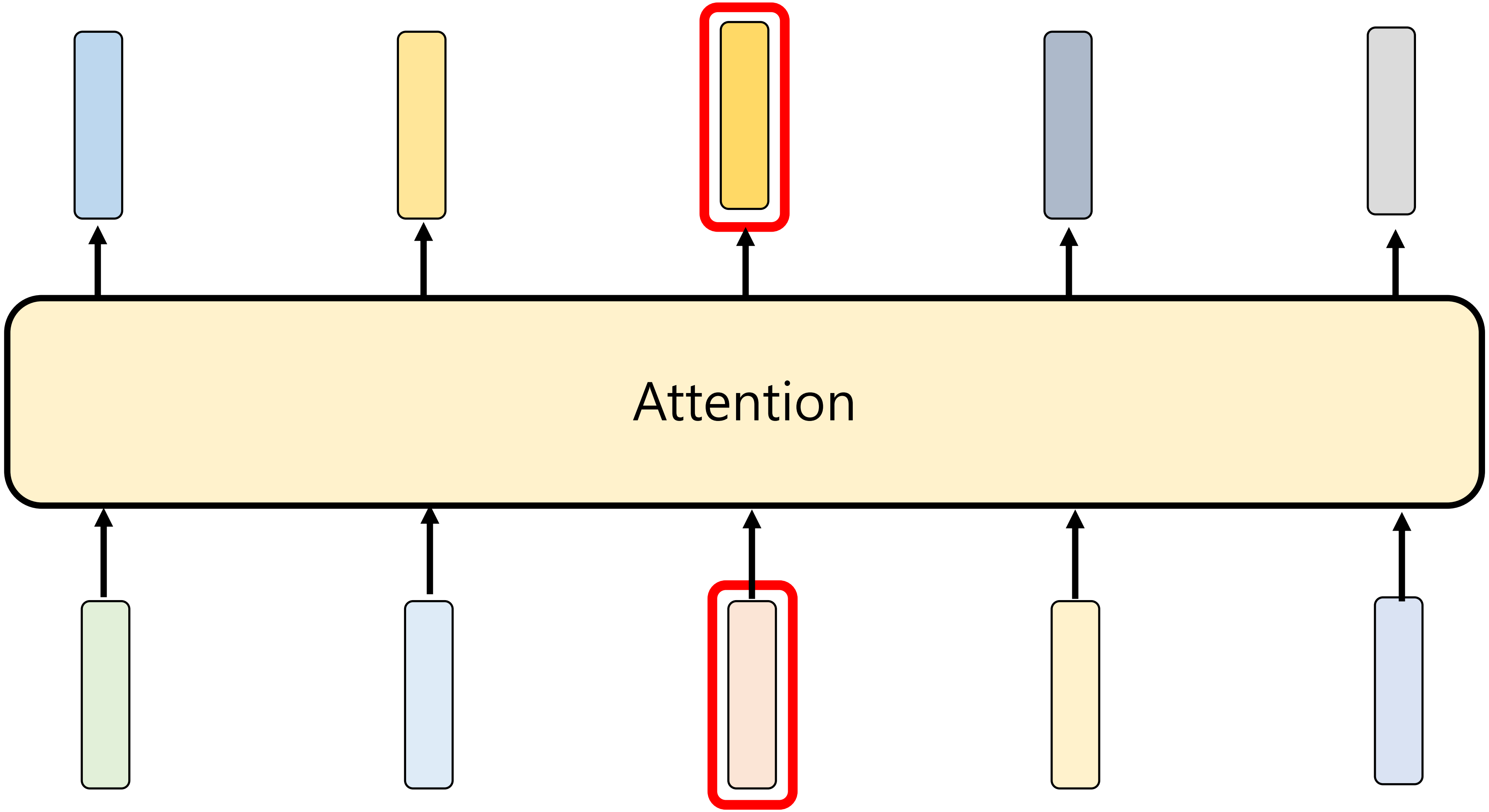
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

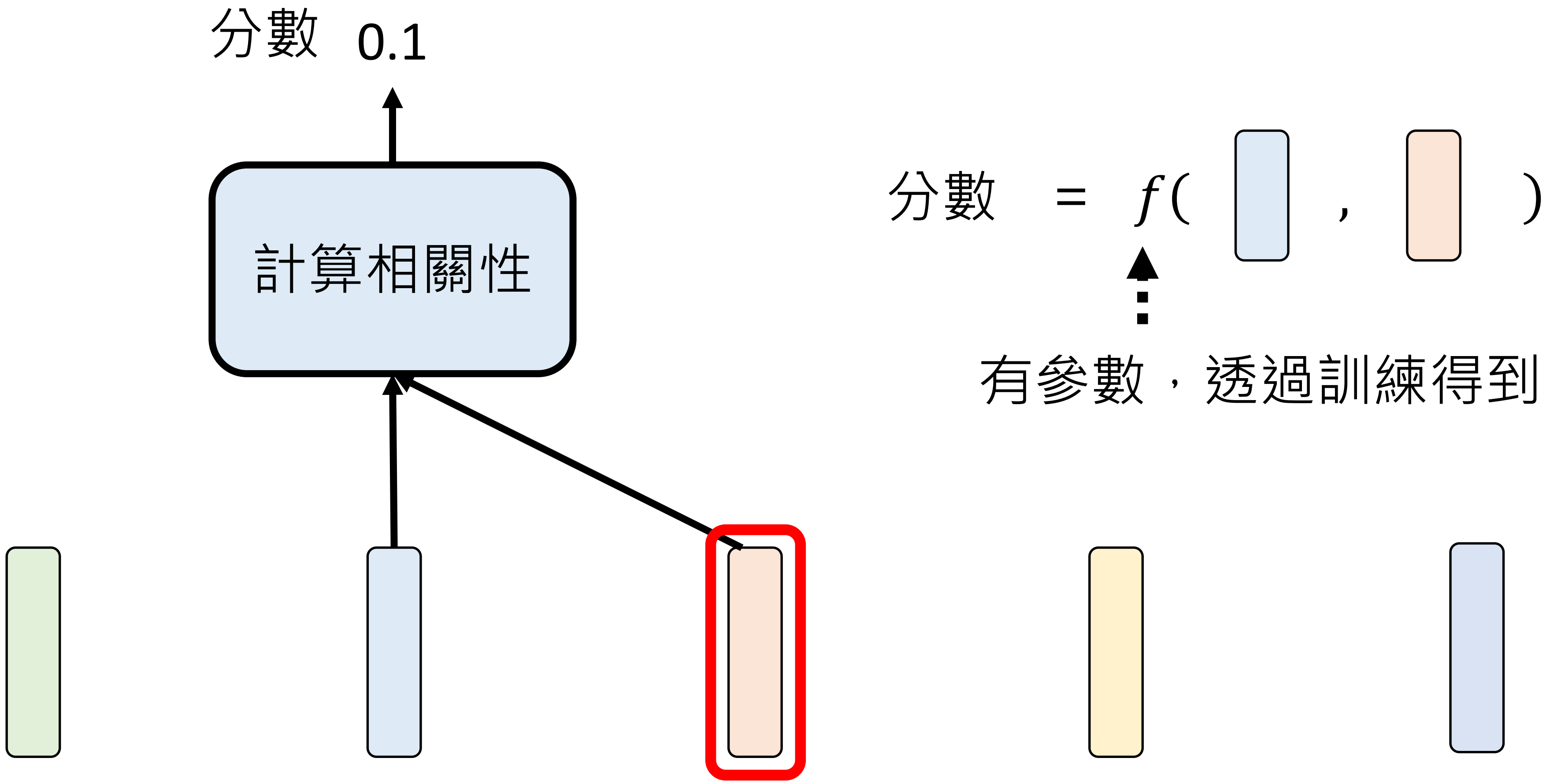
這篇文章主要的貢獻是發現不需要 Recurrent Neural Network (RNN)，只需要 Attention 就夠了

3. Attention: 考慮上下文



3. Attention: 考慮上下文

找出相關的 Token



分數 0.1

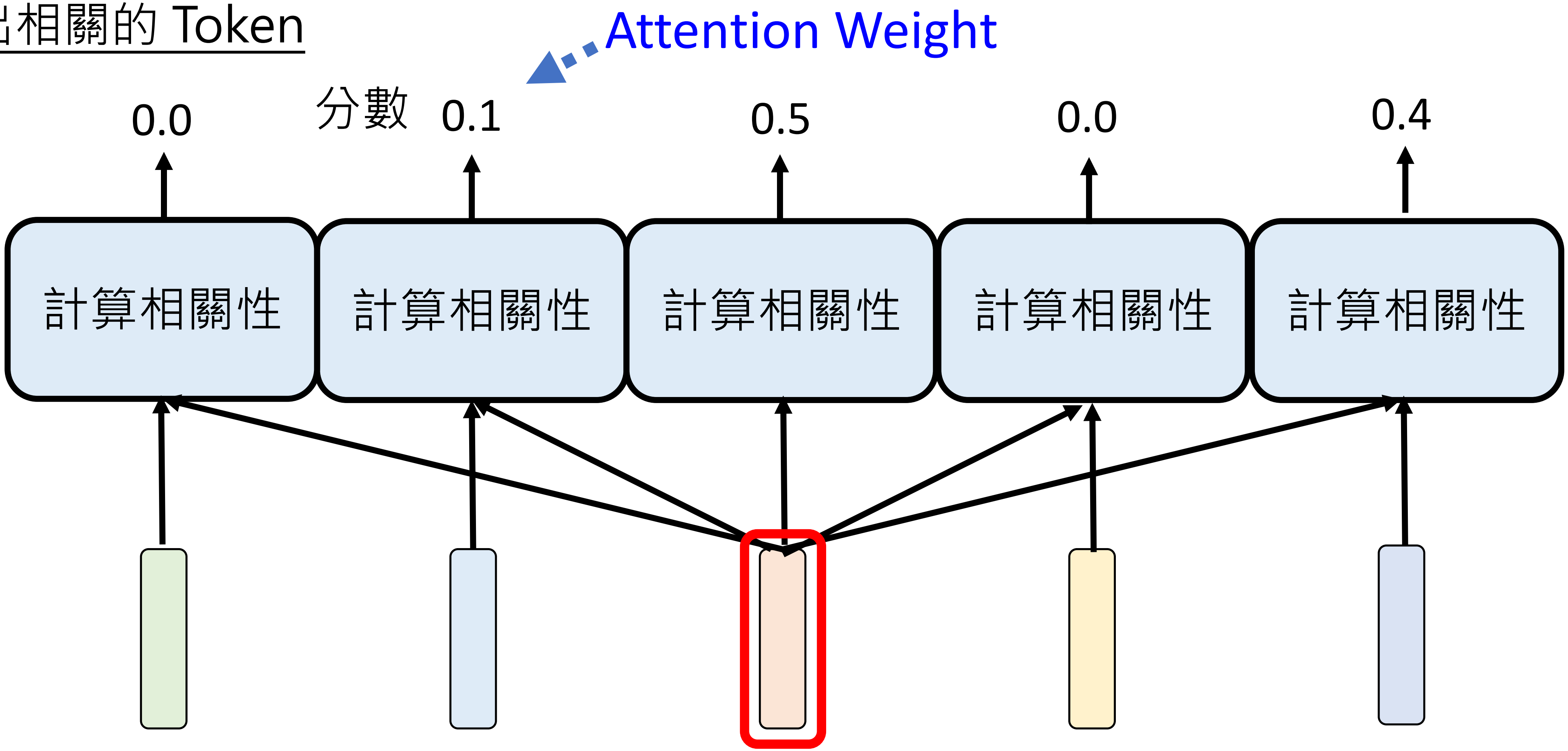
計算相關性

分數 = f(▯ , ▯)

有參數，透過訓練得到

3. Attention: 考慮上下文

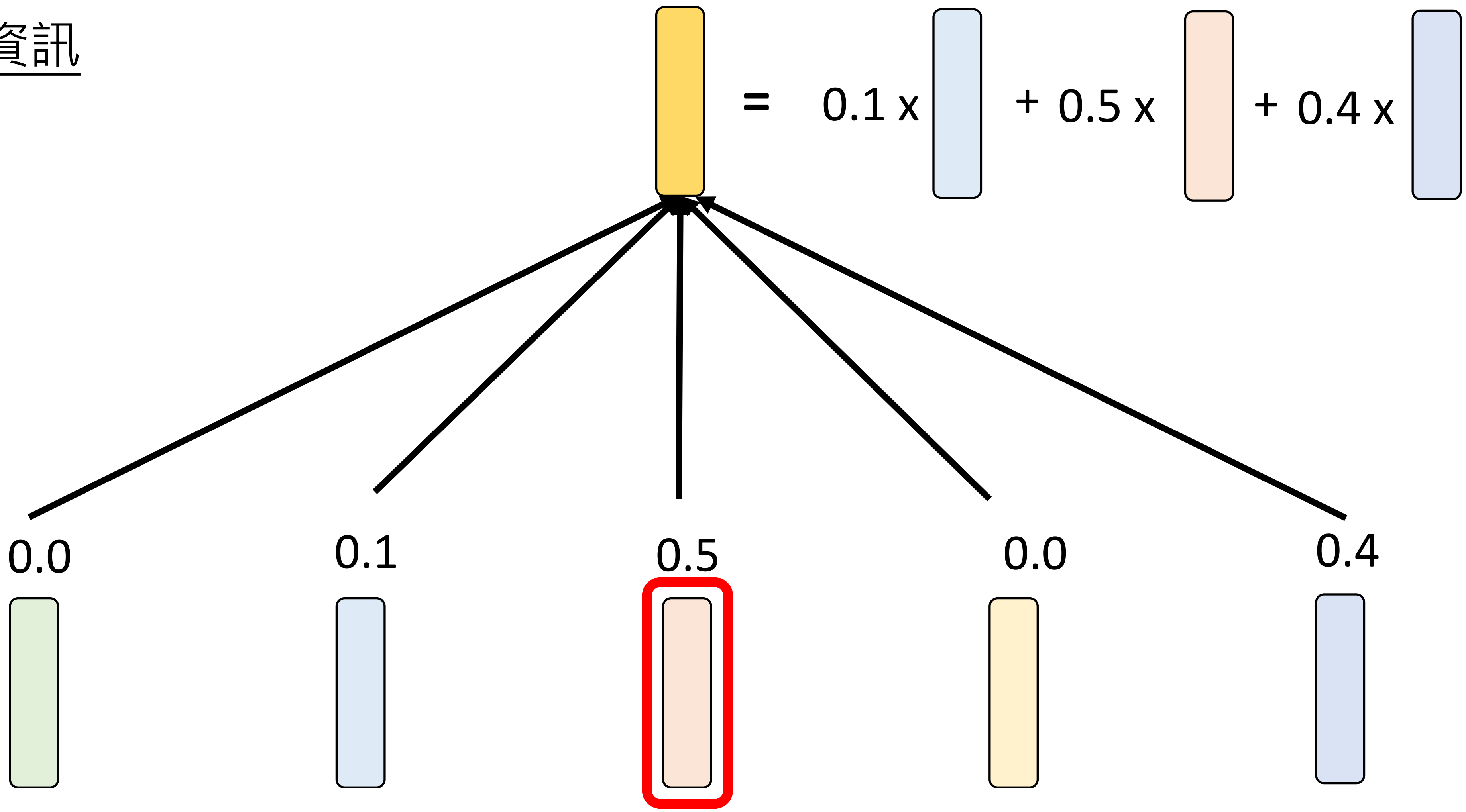
找出相關的 Token



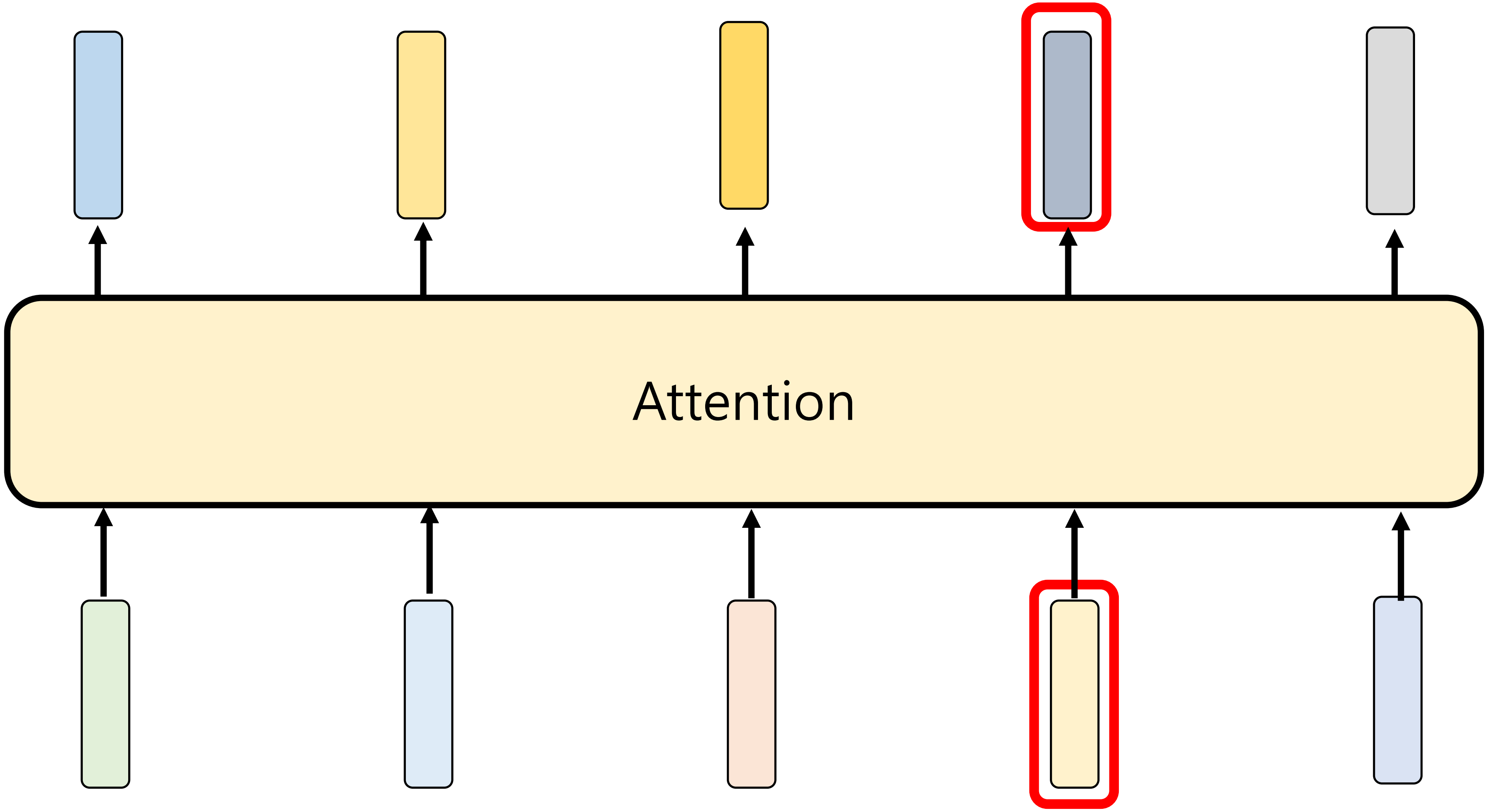
(此處為簡化的講法，事實上在相加前還要過一個 Linear Transformer)

3. Attention: 考慮上下文

集合相關資訊

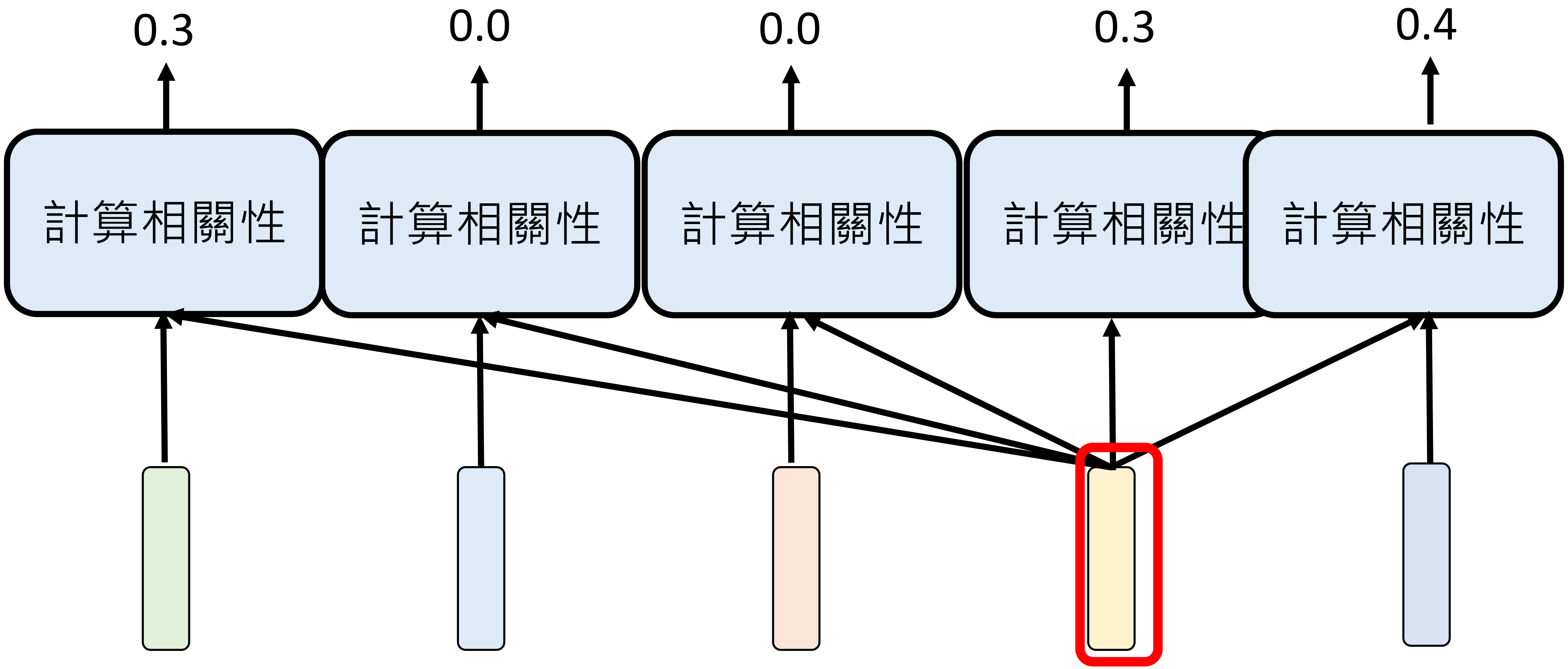


3. Attention: 考慮上下文



3. Attention: 考慮上下文

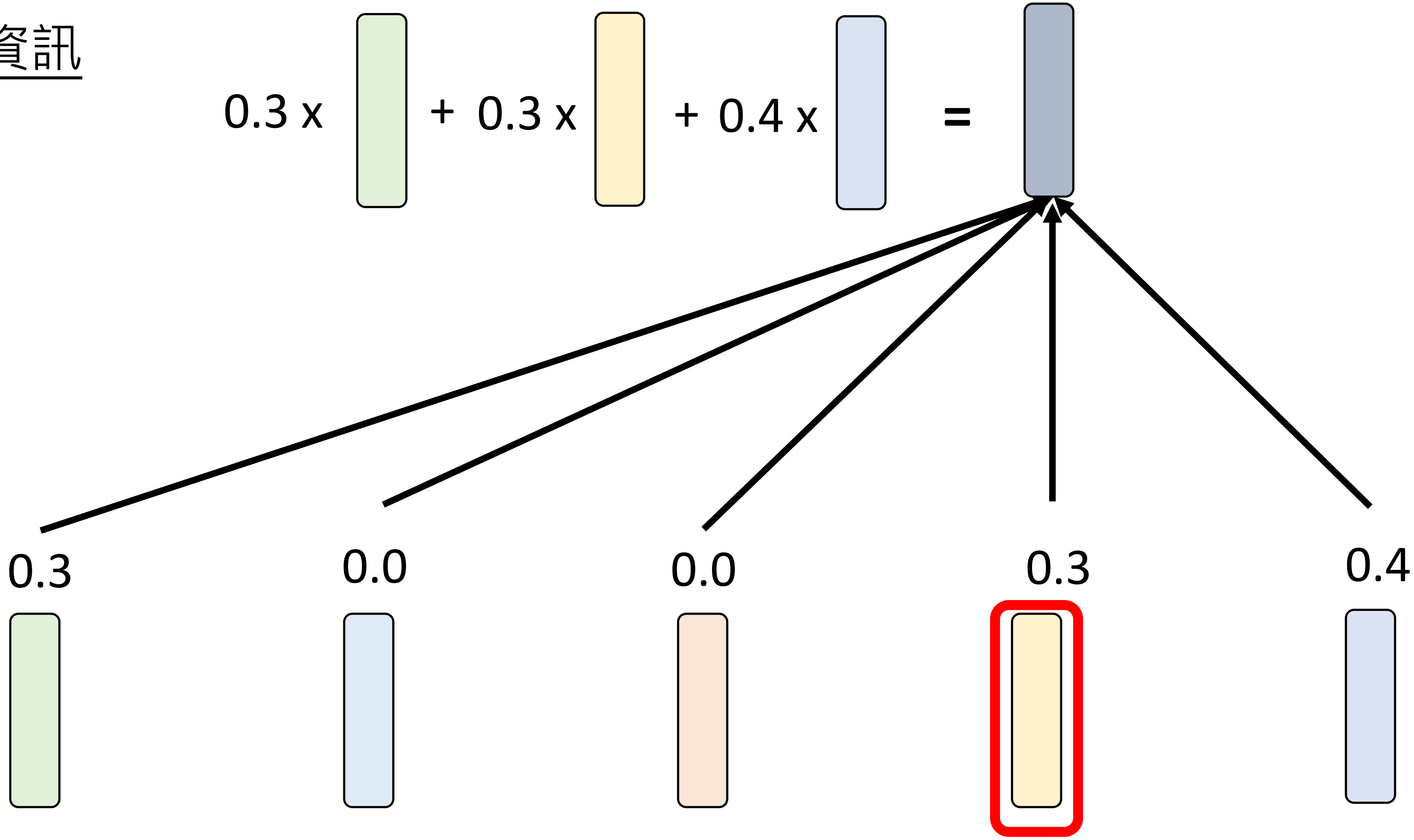
找出相關的 Token



(此處為簡化的講法，事實上在相加前還要過一個 Linear Transformer)

3. Attention: 考慮上下文

集合相關資訊



3. Attention: 考慮上下文

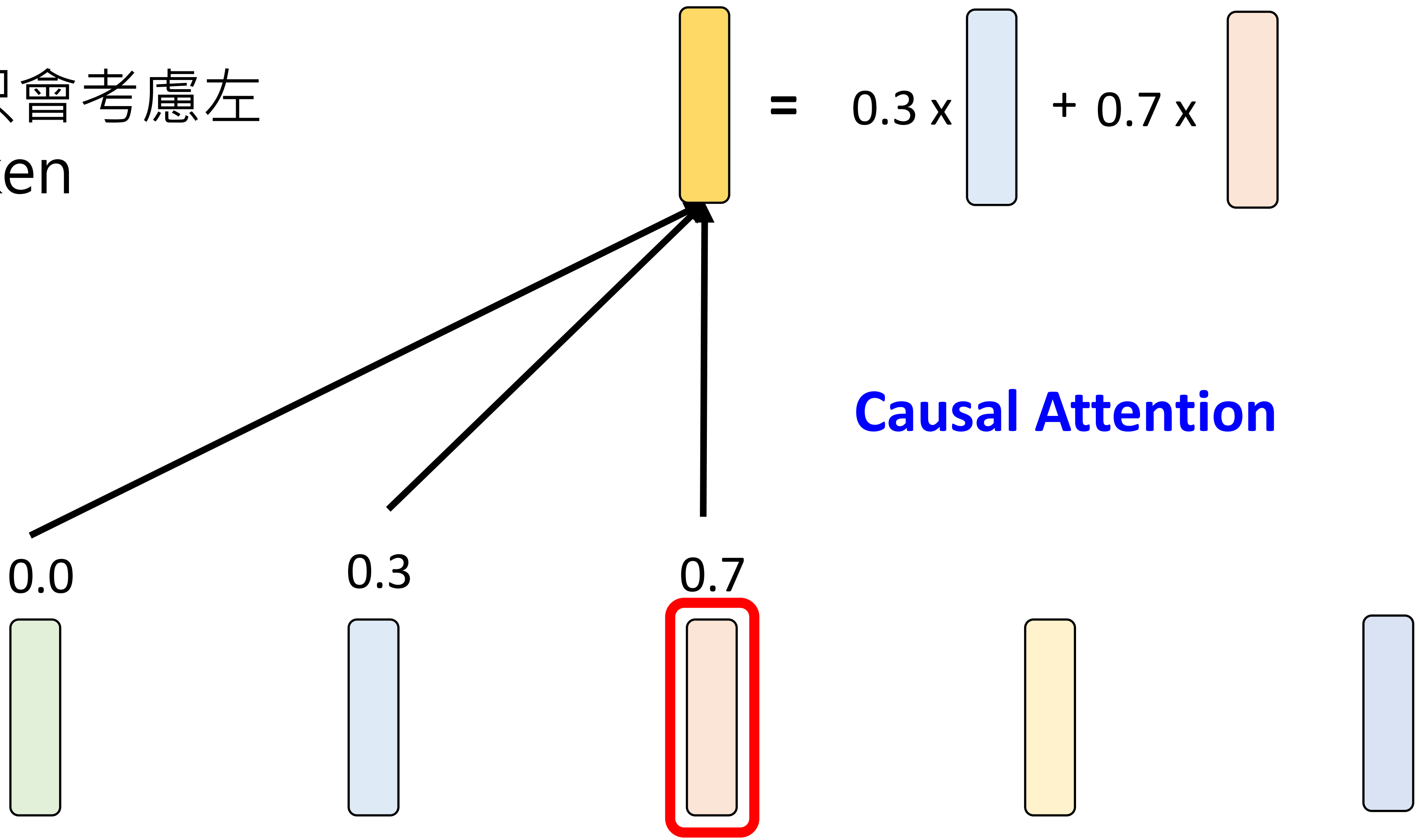
Attention Matrix

	🟩	🟦	🟧	🟨	🟦
🟩	0.5	0.0	0.0	0.3	0.1
🟦	0.5	0.9	0.1	0.0	0.2
🟧	0.0	0.0	0.5	0.0	0.2
🟨	0.0	0.0	0.0	0.3	0.2
🟦	0.0	0.4	0.4	0.4	0.2

計算所有 Token 兩兩間的相關性

3. Attention: 考慮上下文

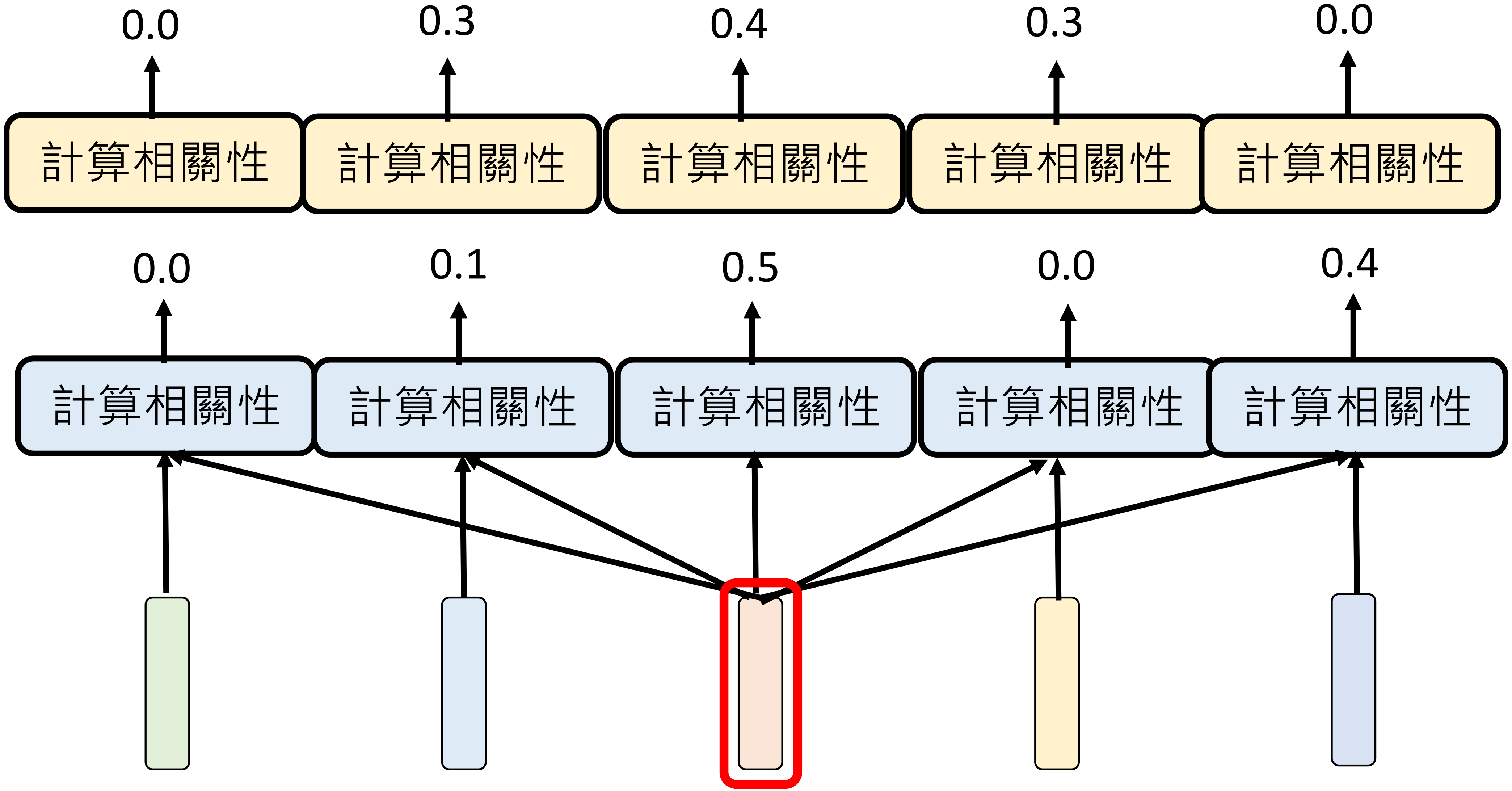
實作時只會考慮左邊的 token



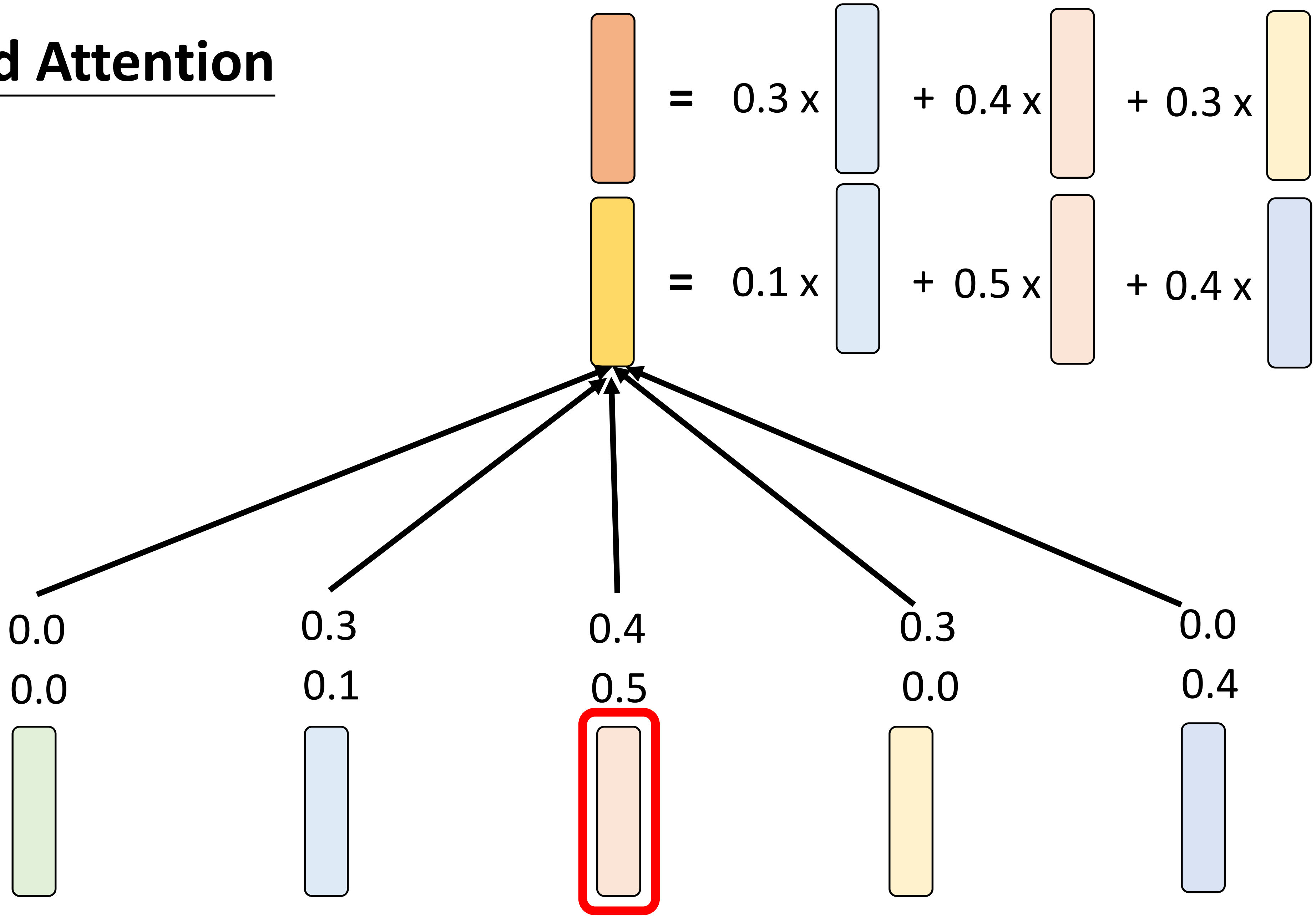
Causal Attention

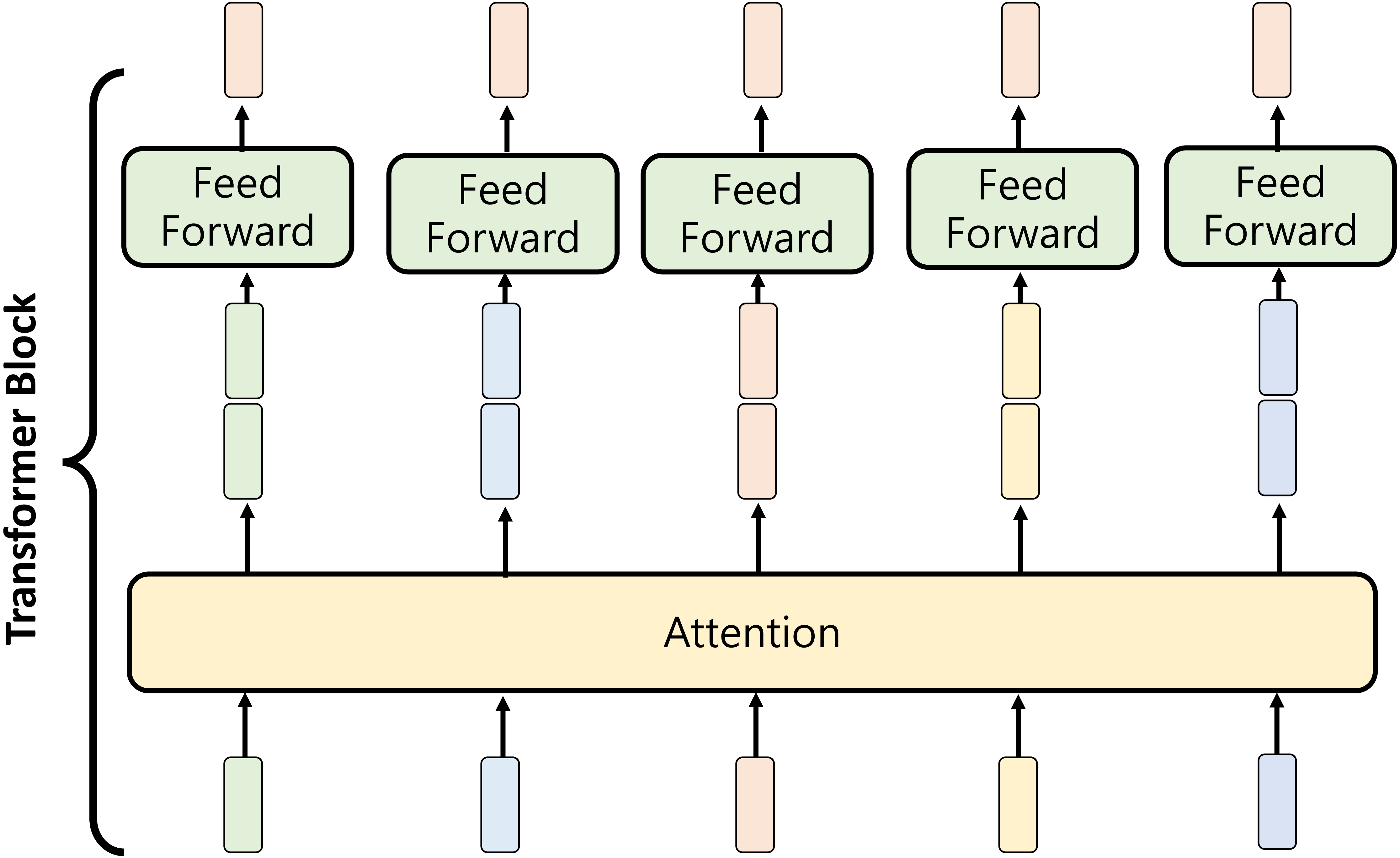
Multi-head Attention

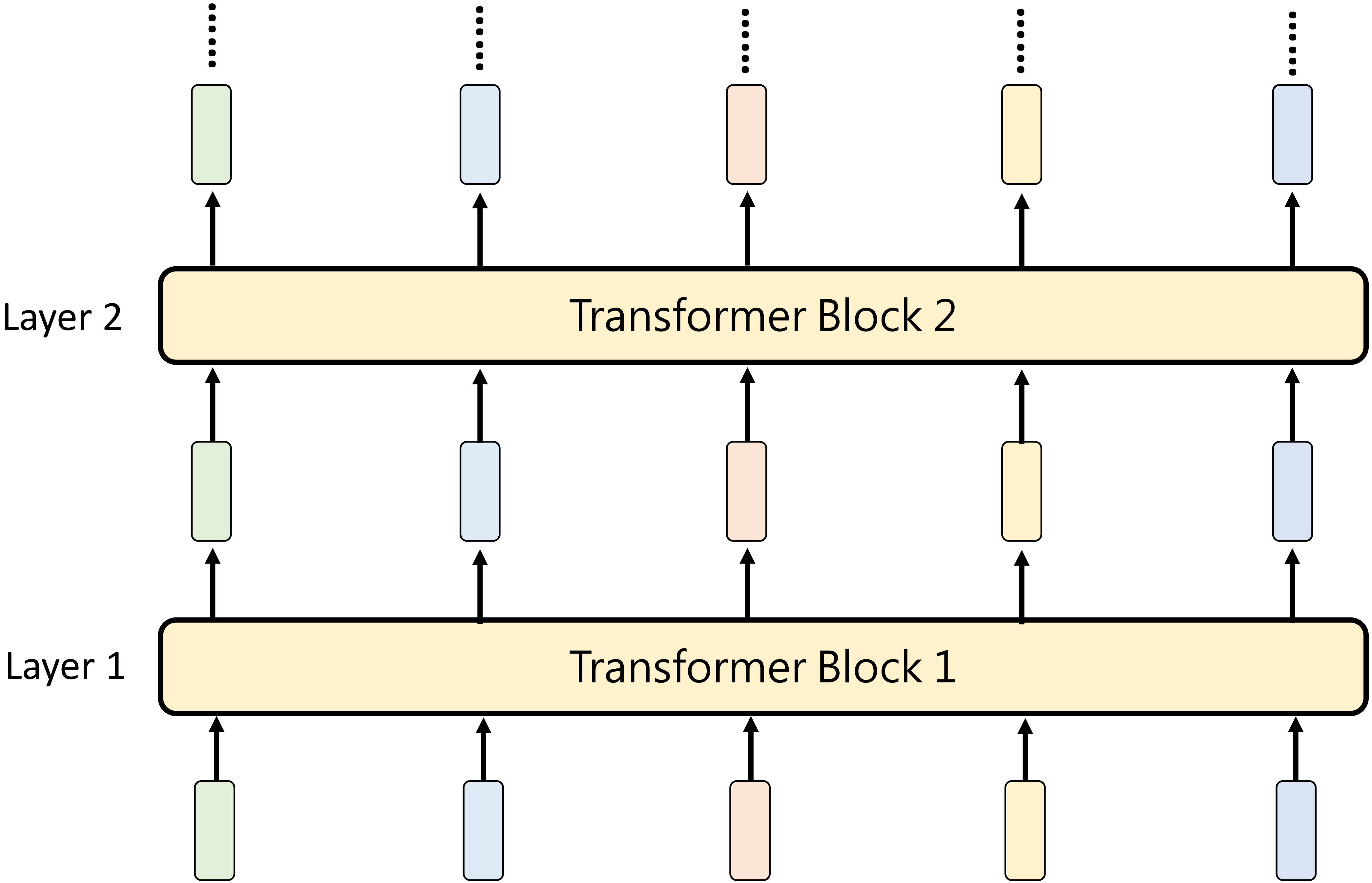
關聯性不只一種



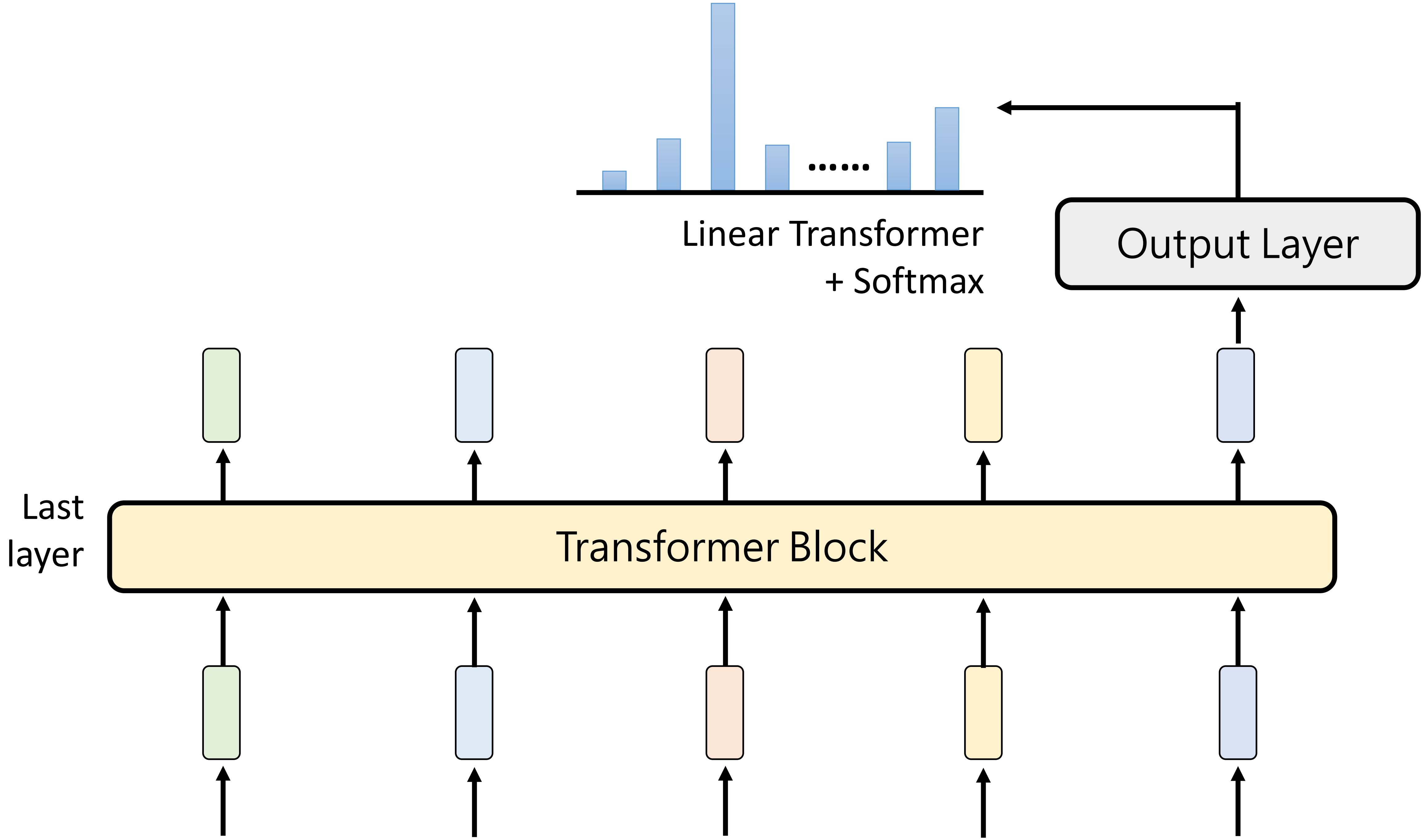
Multi-head Attention



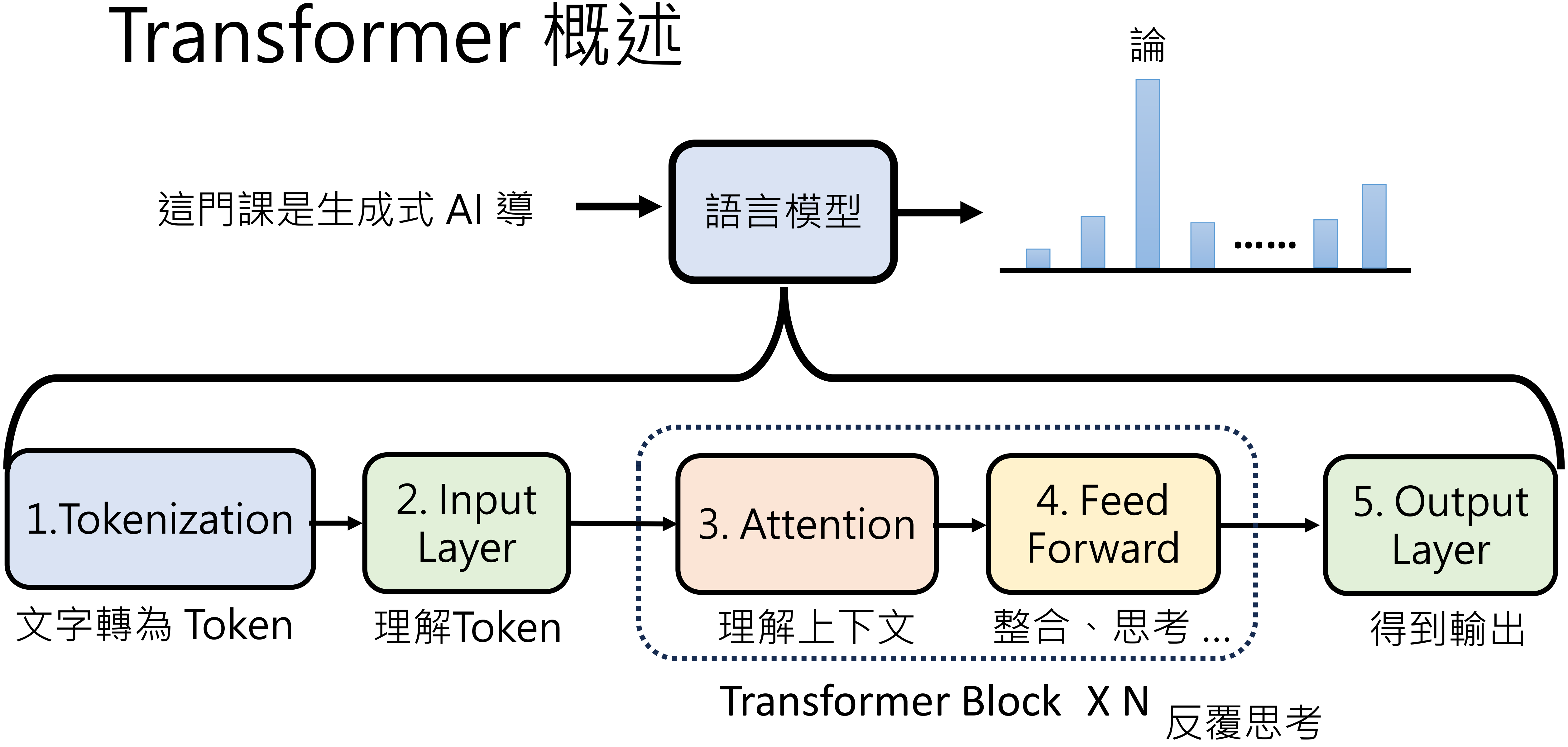

Transformer Block
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Attention



Layer 2
Transformer Block 2
Layer 1
Transformer Block 1

Linear Transformer
+ Softmax
Output Layer
Last
layer
Transformer Block

Transformer 概述


論
這門課是生成式 AI 導
語言模型
......
1.Tokenization
文字轉為 Token
2. Input Layer
理解Token
3. Attention
理解上下文
4. Feed Forward
整合、思考 ...
5. Output Layer
得到輸出
Transformer Block X N
反覆思考