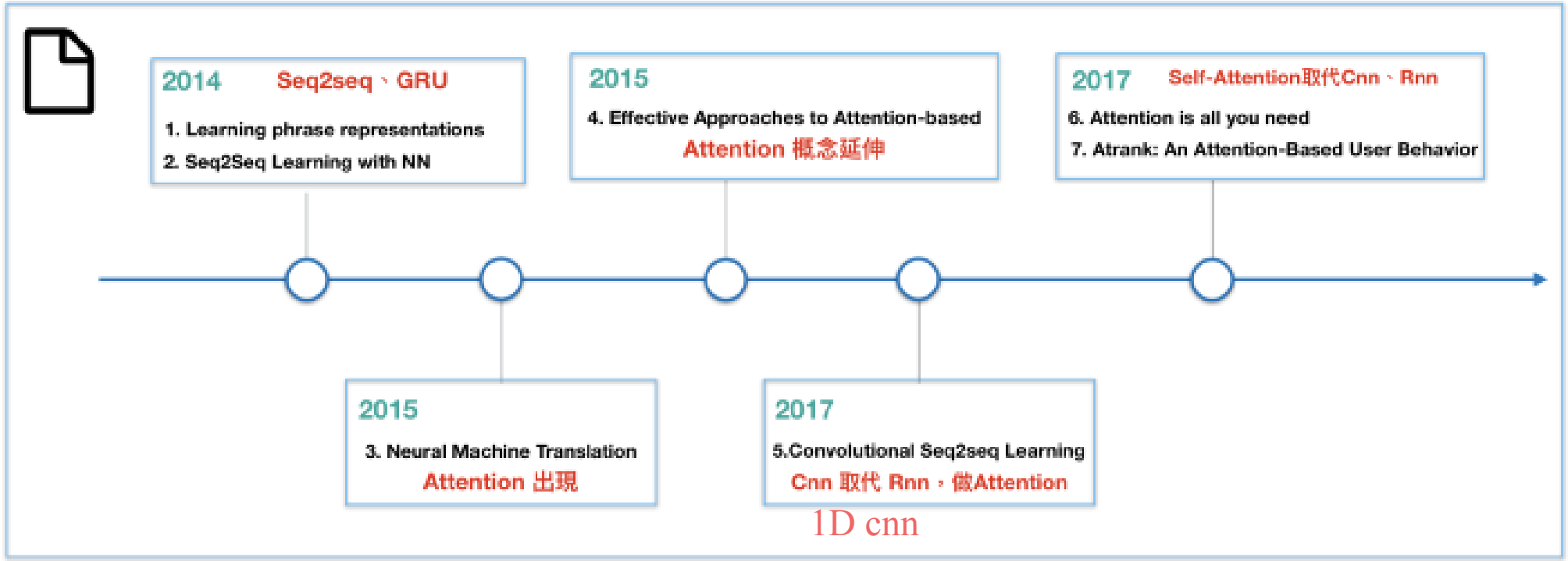
Part.3

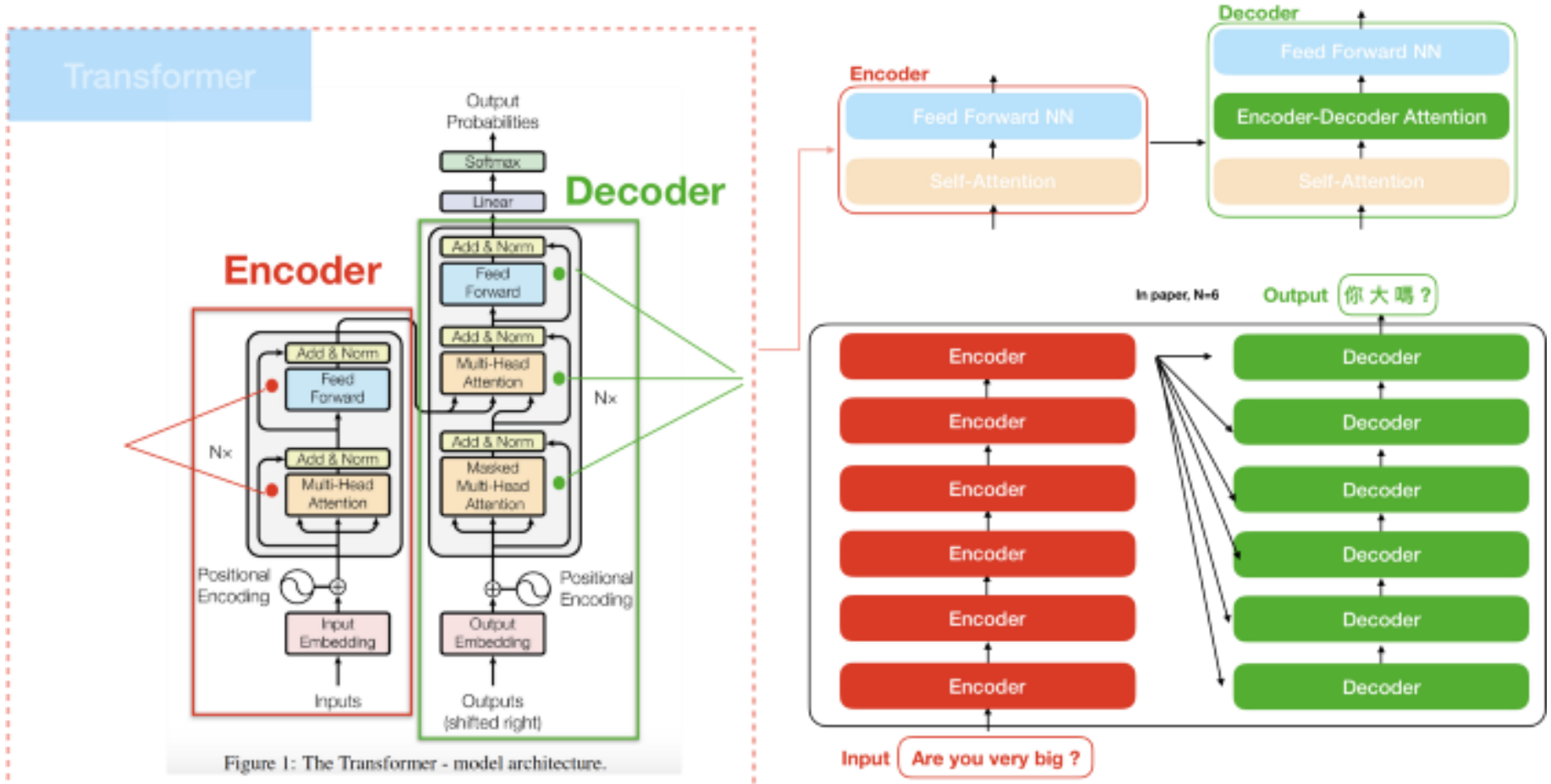
Transformer & BERT

History



https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-%E4%B8%AD%E6%96%87%E7%89%88-ef2ddf8597a4

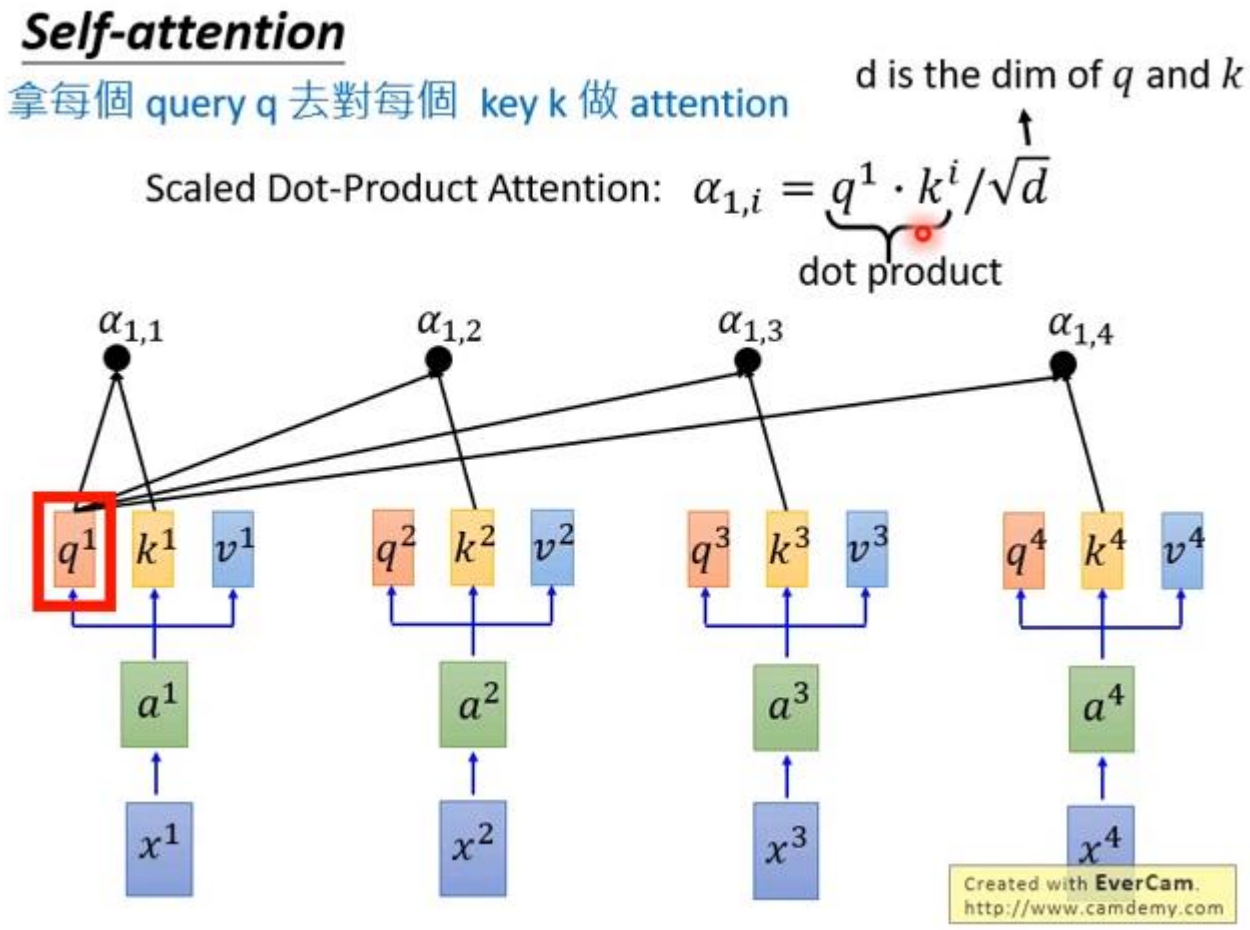
Transformer



https://jalammar.github.io/illustrated-transformer/

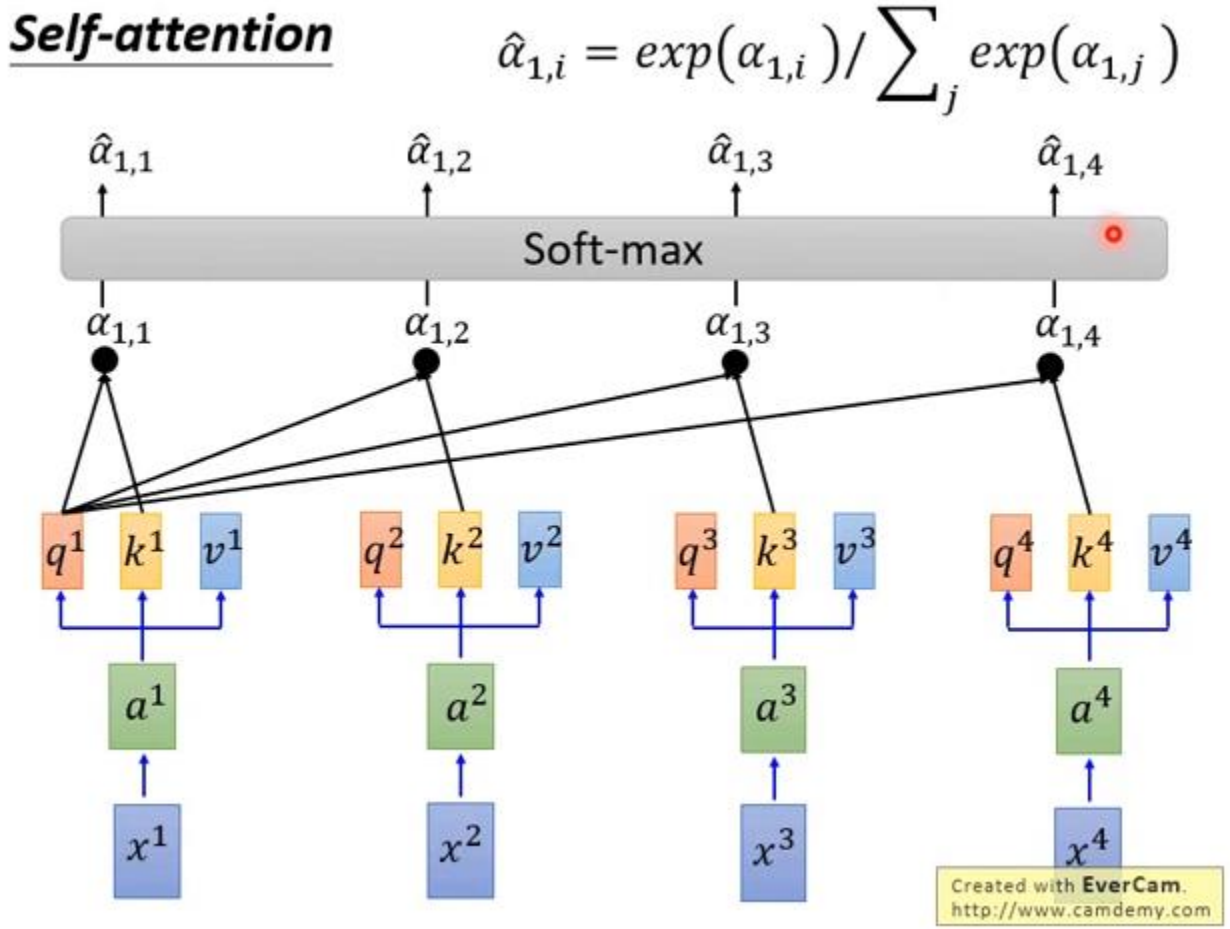
Scaled Dot-Product Attention

- **表達Q跟K的匹配程度**



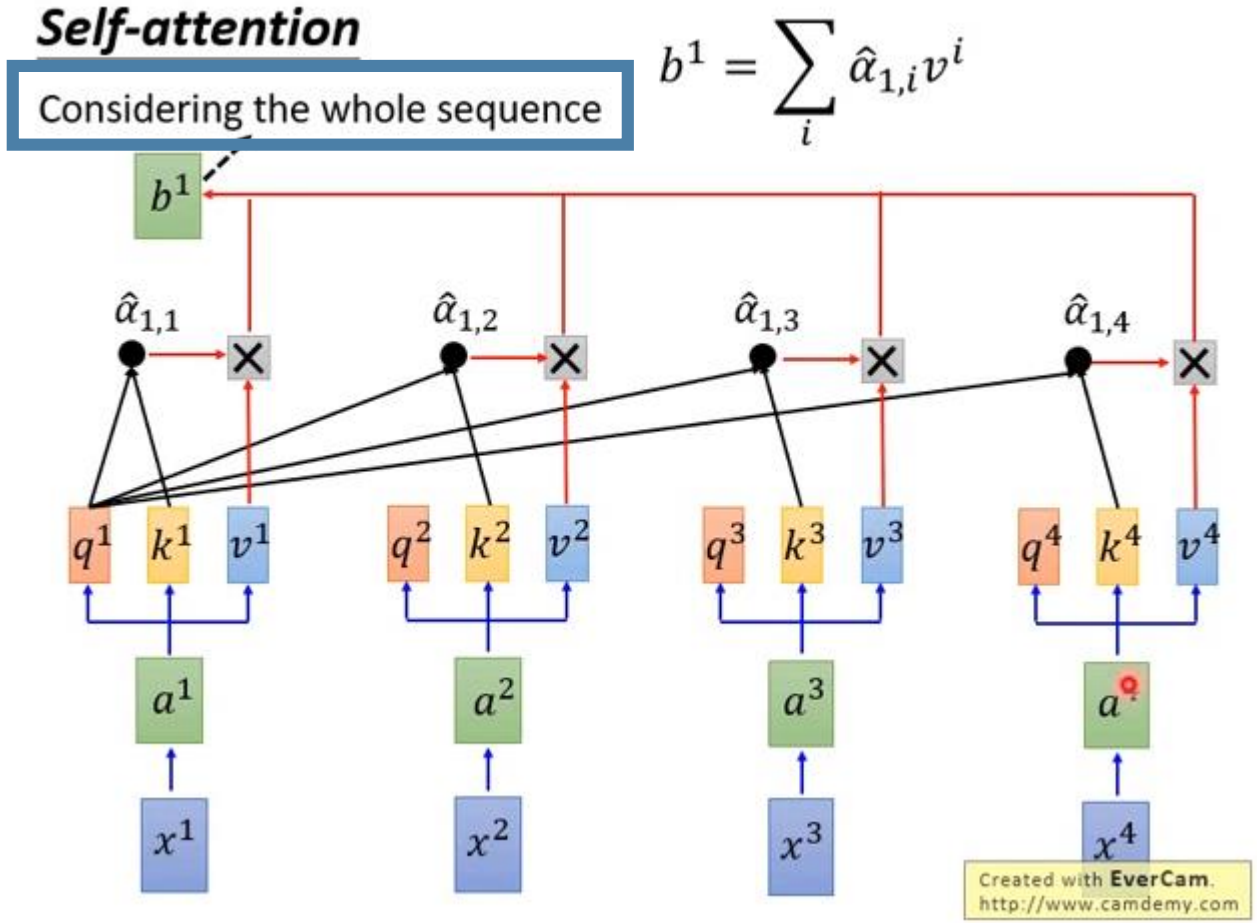
Scaled Dot-Product Attention

- **取softmax得到 Attention score**



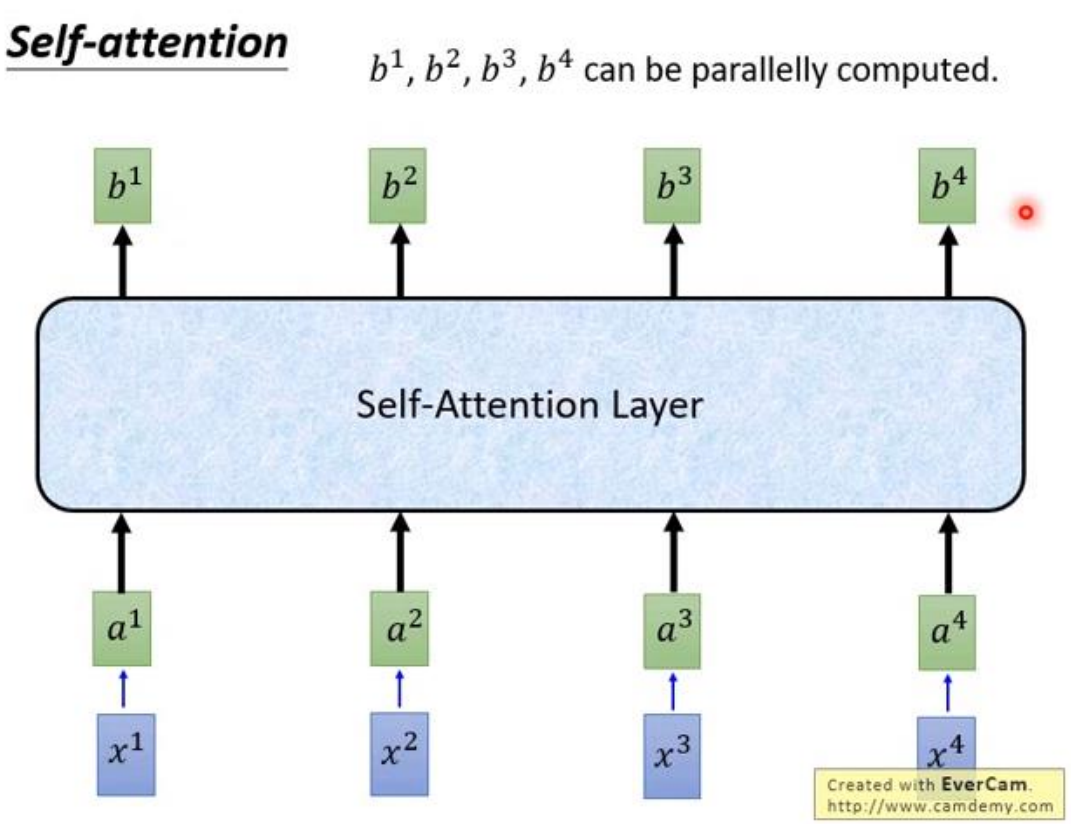
Scaled Dot-Product Attention

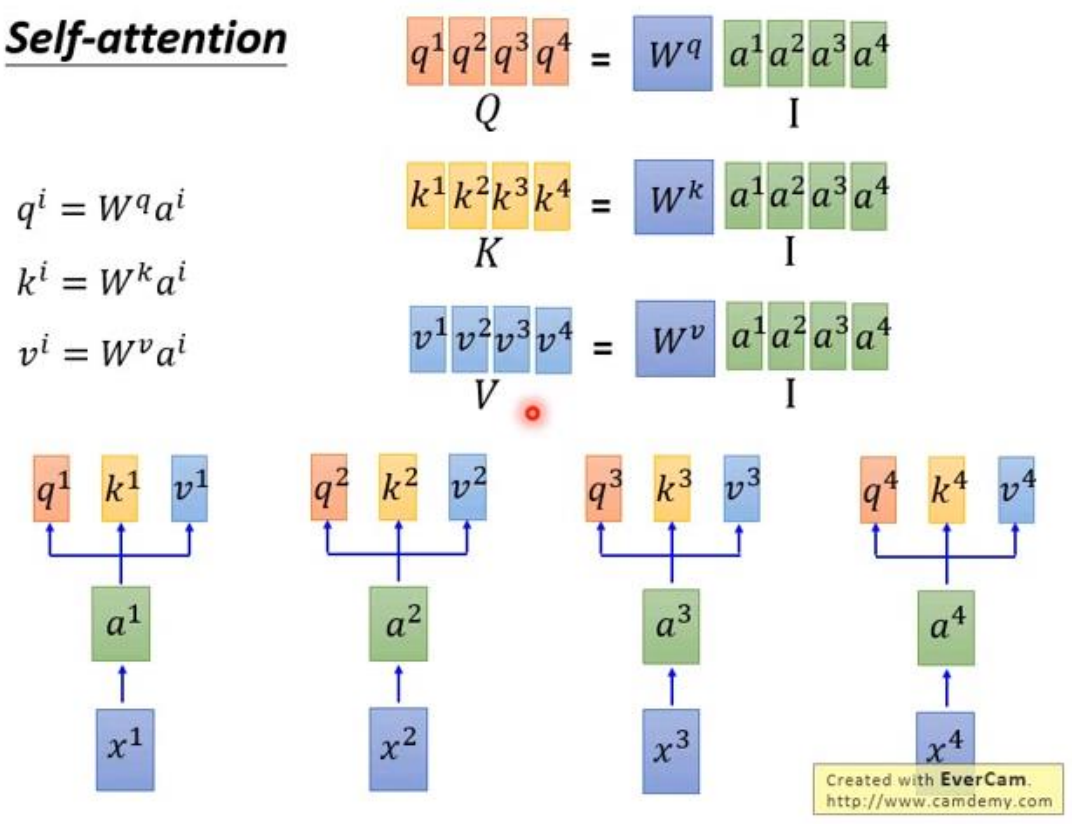
- **Weighted sum**



Scaled Dot-Product Attention

- **平行處理 (矩陣運算)**



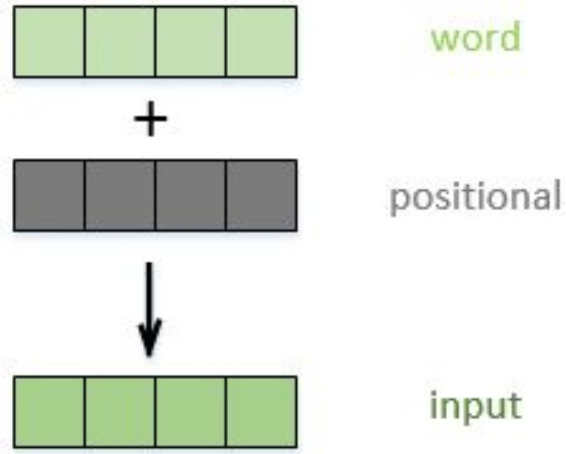


Positional Encoding

- **Self-attention沒有考慮先後順序**
- **額外加入位置的訊息**
- **同時考慮語意和詞在句字中的位置**
- **PE公式:**

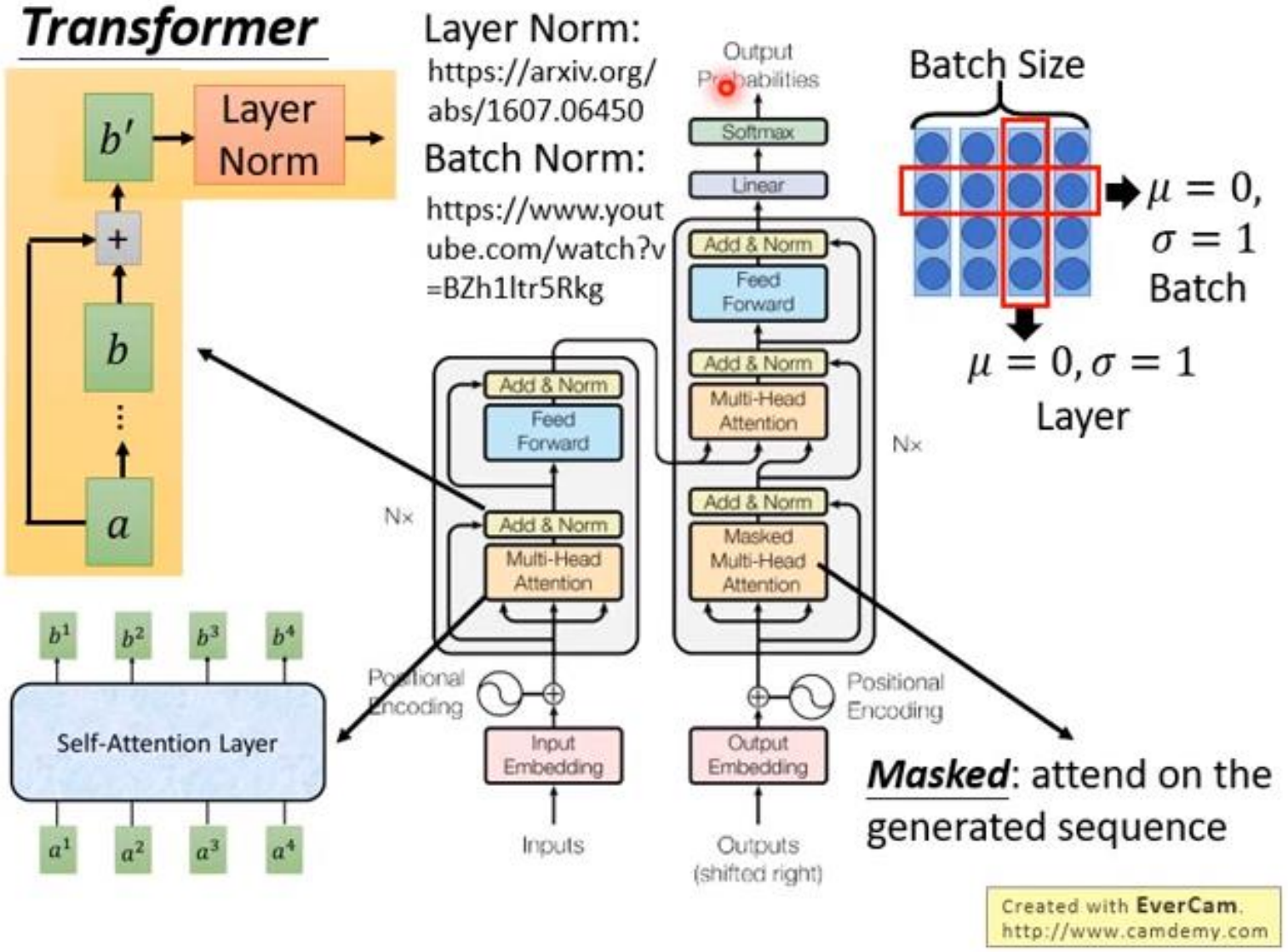
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$



https://leemeng.tw/neural-machine-translation-with-transformer-and-tensorflow2.html

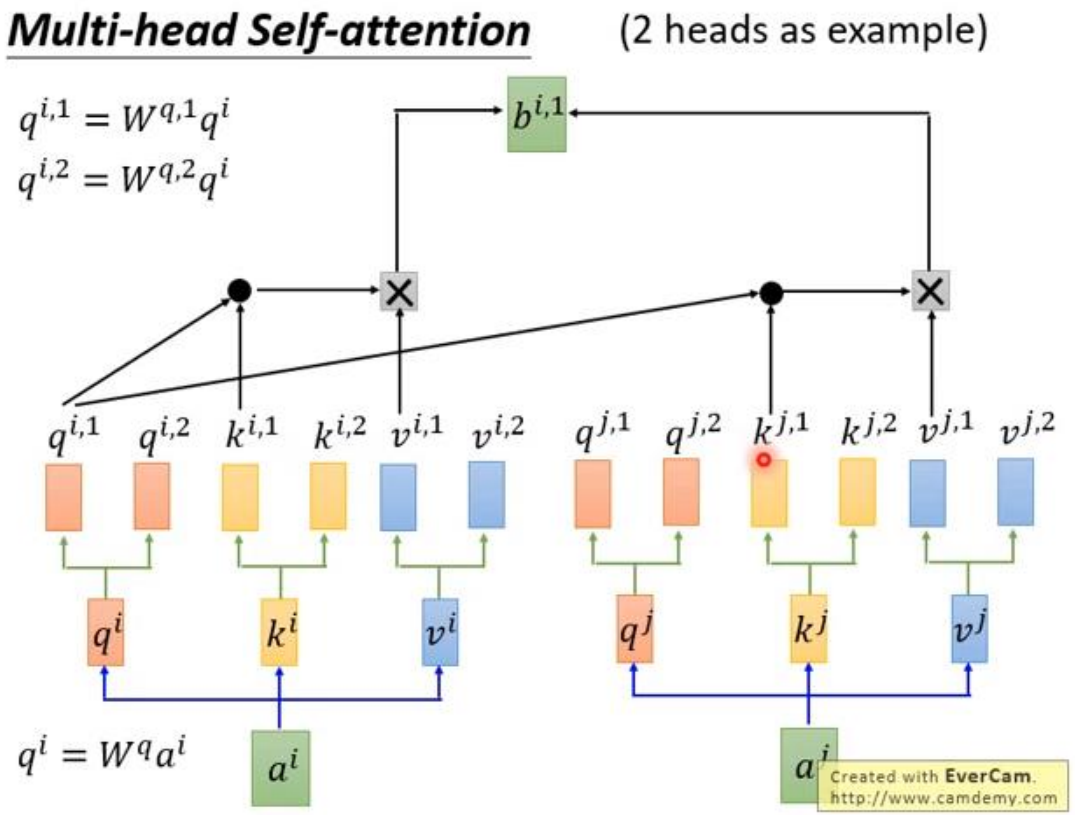
Transformer

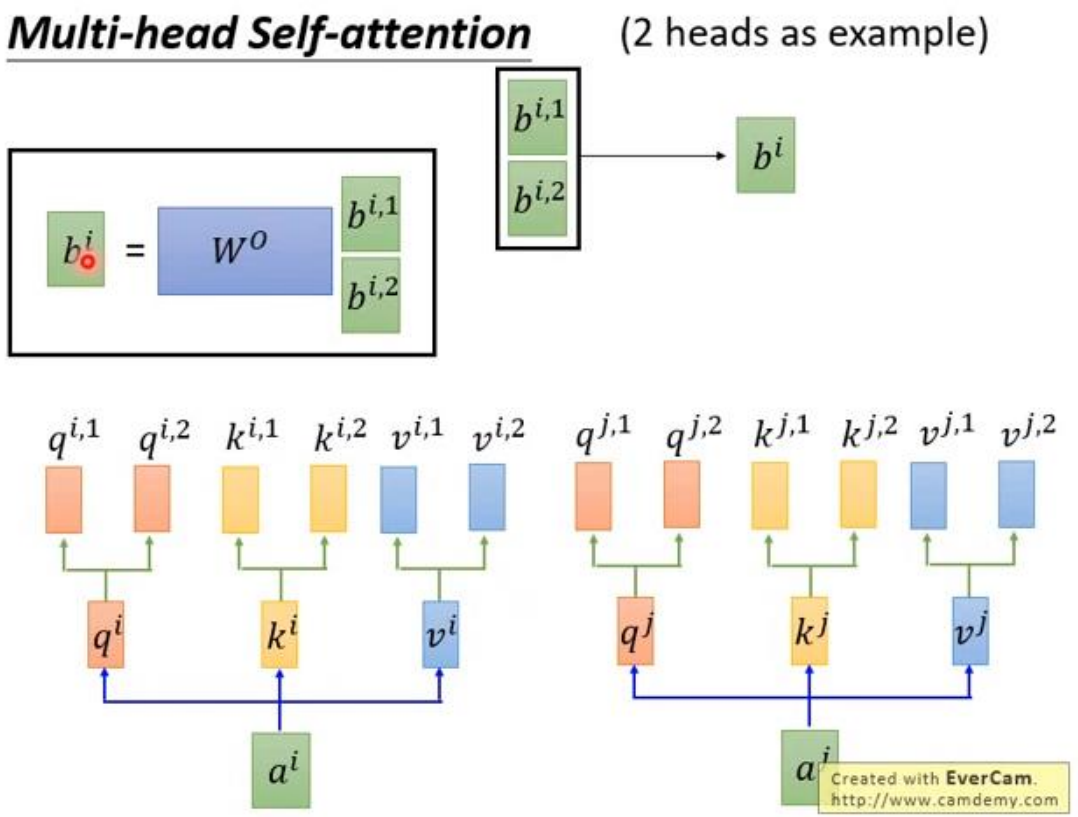


Multi-Head Attention

Multi-Head Attention

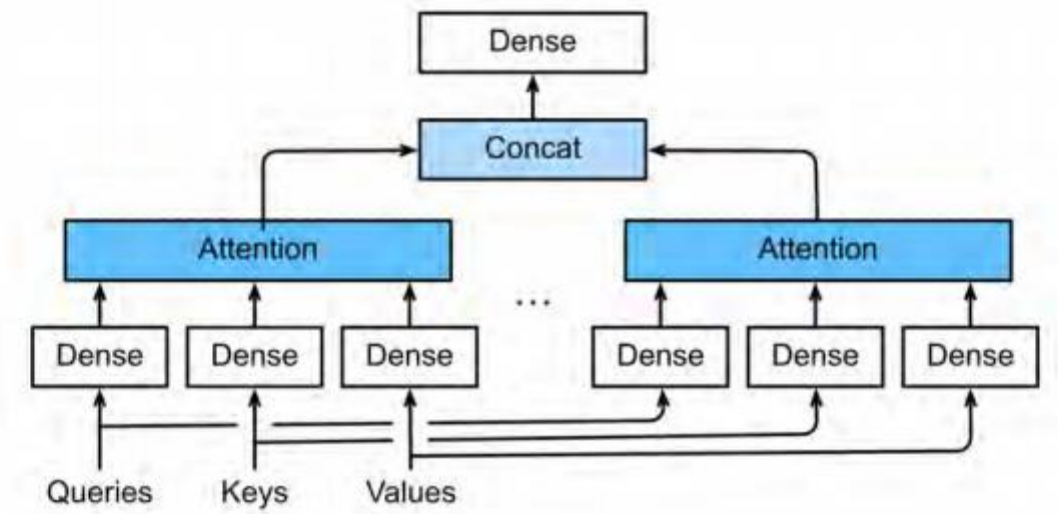
- **分裂q, k, v (head能各自關注不同重點)**





Multi-Head Attention

Q: query
K: key
V: value



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots \text{head}_h)W^O$$

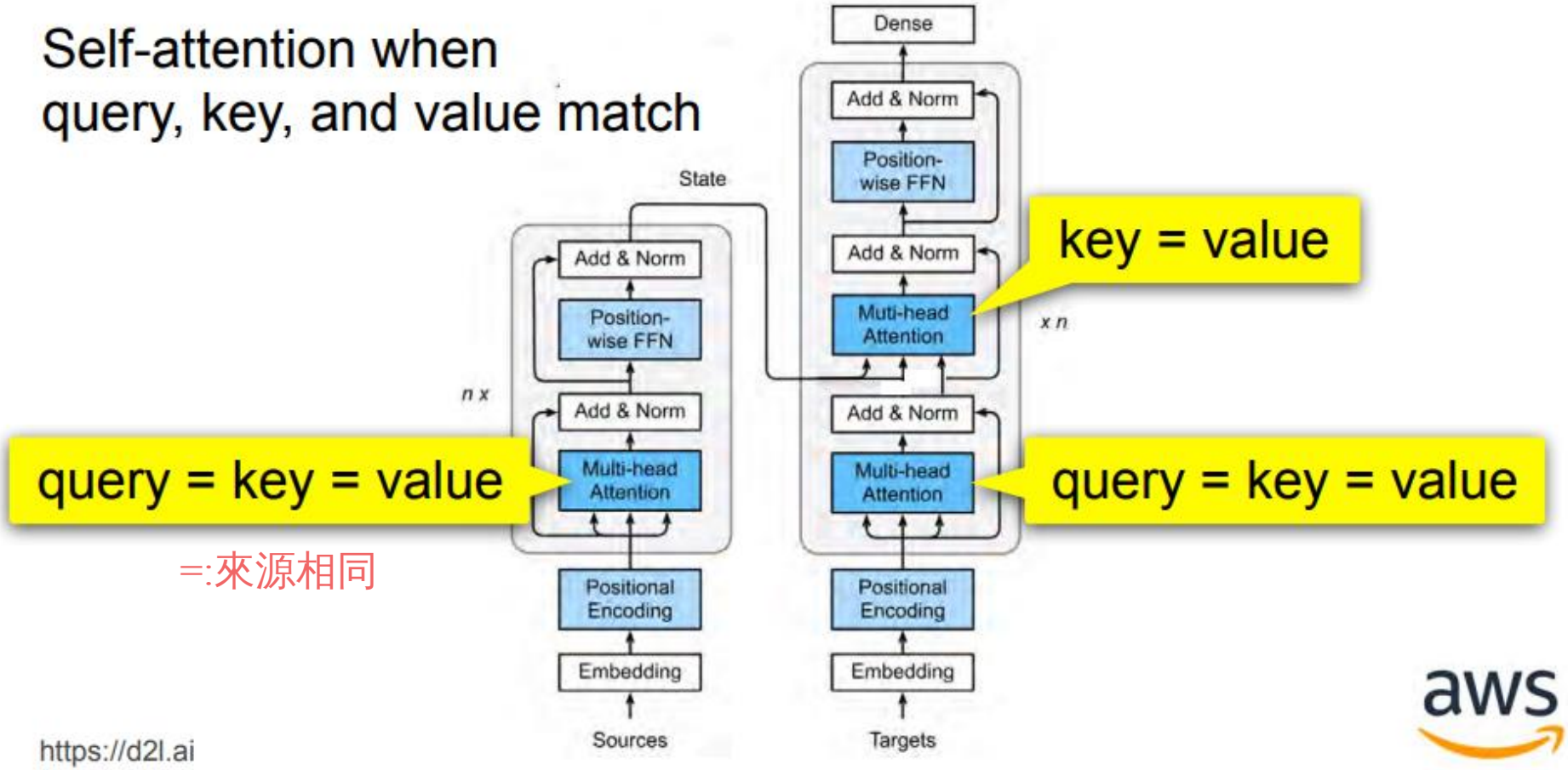
$$\text{where head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

https://d2l.ai

aws

Multi-Head Attention

Self-attention when
query, key, and value match



=:來源相同

https://d2l.ai

aws

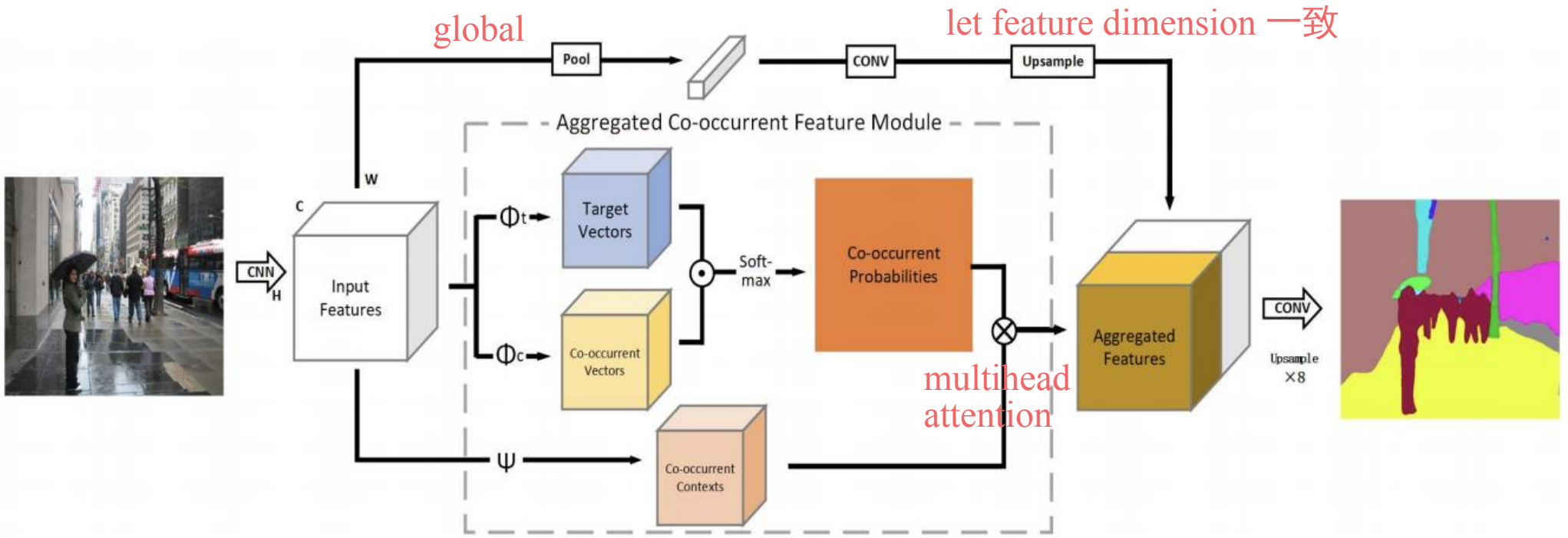
Semantic Segmentation

Semantic Segmentation

Semantic Segmentation

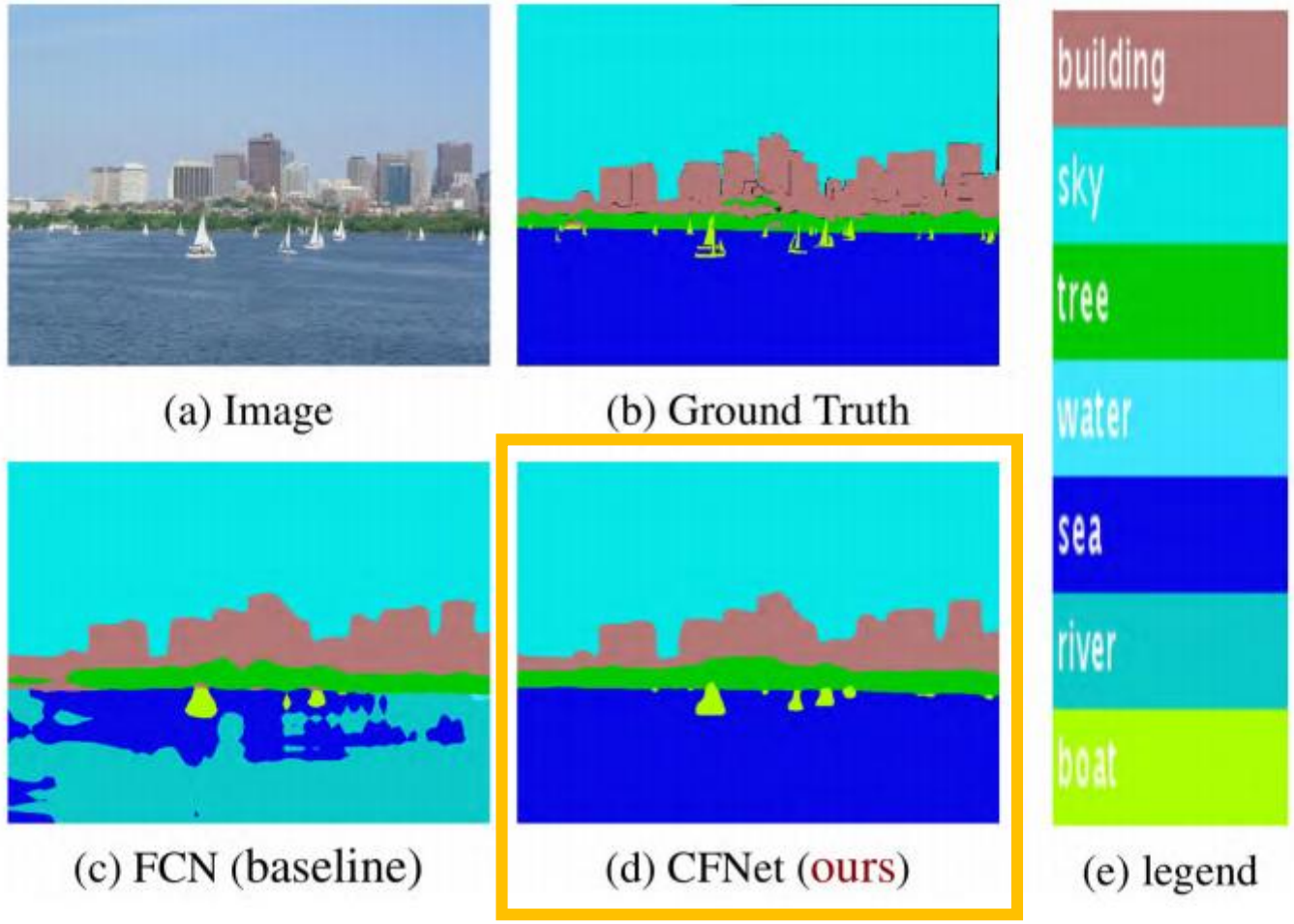
Co-occurrent Features

Multi-head attention for semantic segmentation (Zhang et al., '19)



Multi-Head Attention

Classify pixels co-occurring with boat as sea rather than water



https://d2l.ai

Q & A

- 回顧一下Transformer是甚麼?
- 把Attention換成Self Attention有什麼好處?

BERT

BERT
Bidirectional Encoder Representations from Transformers
(Devlin et al, 2018)

SOTA on 11 NLP tasks

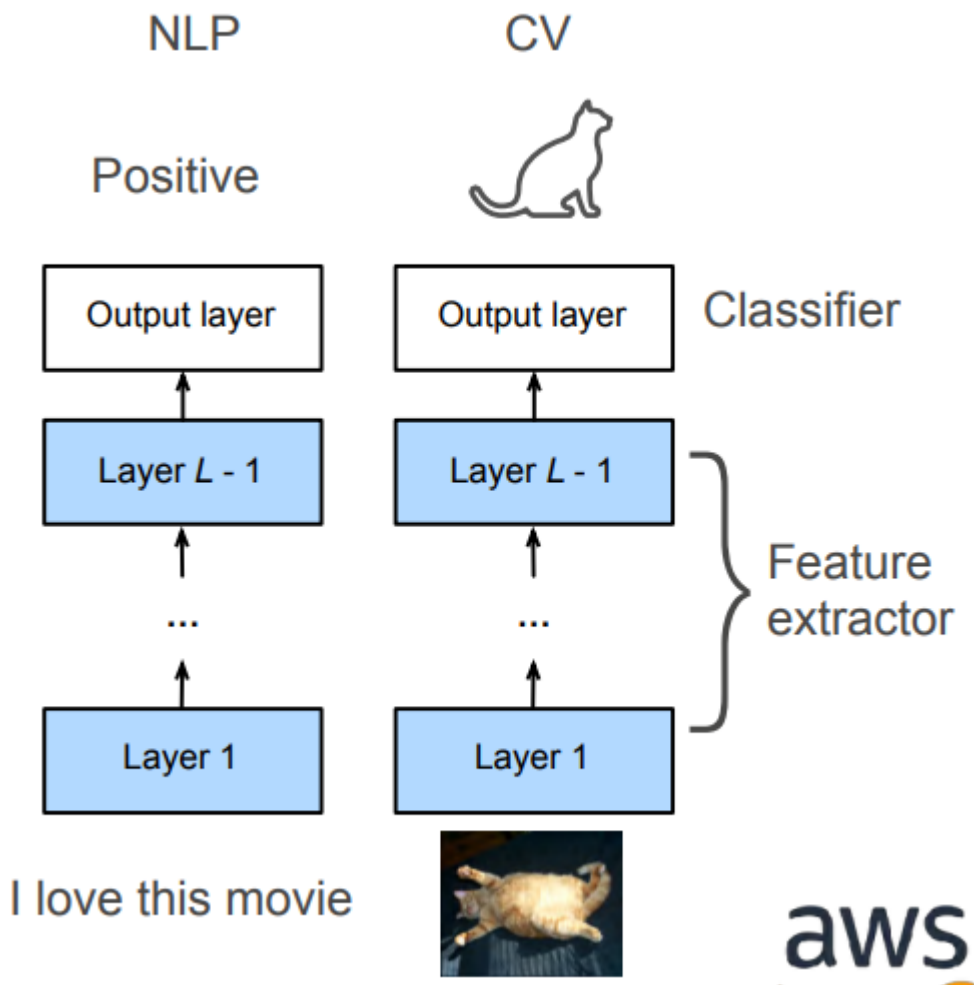
courses.d2l.ai/berkeley-stat-157/index.html

https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html

Motivation of BERT

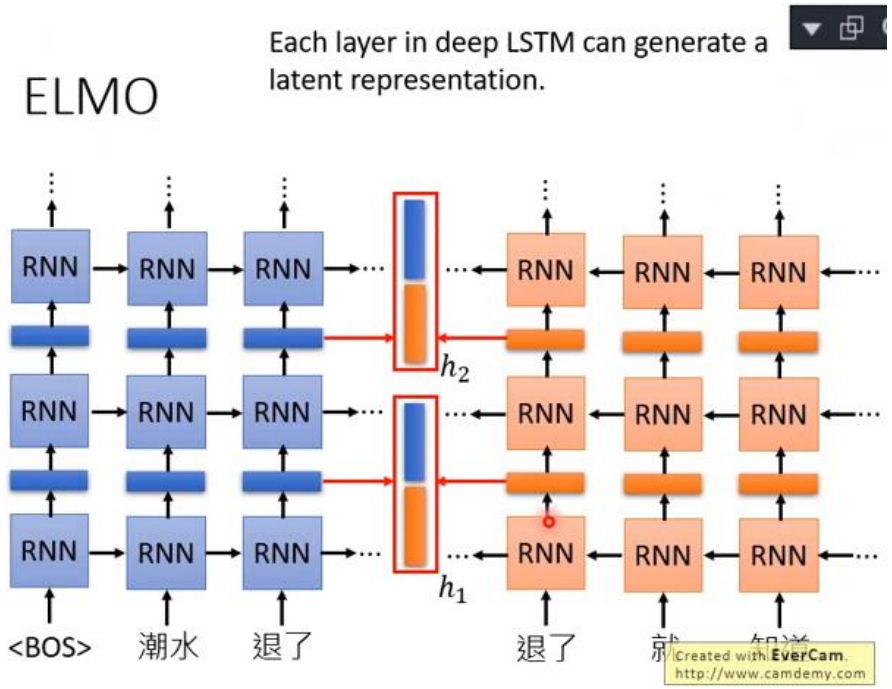
Motivation

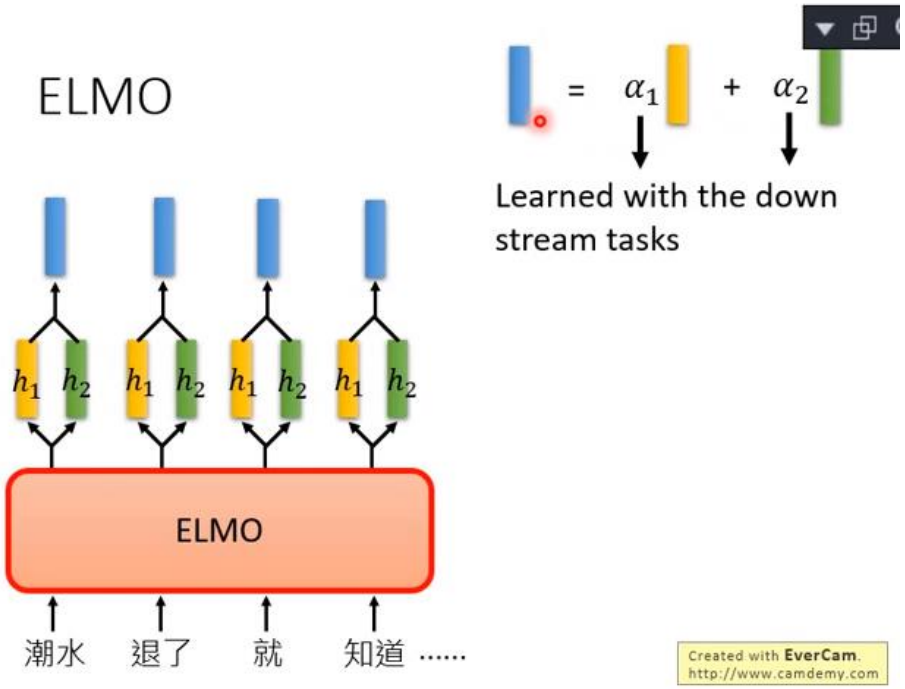
- Fine-tuning for NLP (learning a prior for NLP)
- Pre-trained model captures prior
- Only add one (or more) output layers for new task



ELMO

- **Embeddings from Language Models (為了解決一詞多義)**
- **BERT 前的 Language Model**





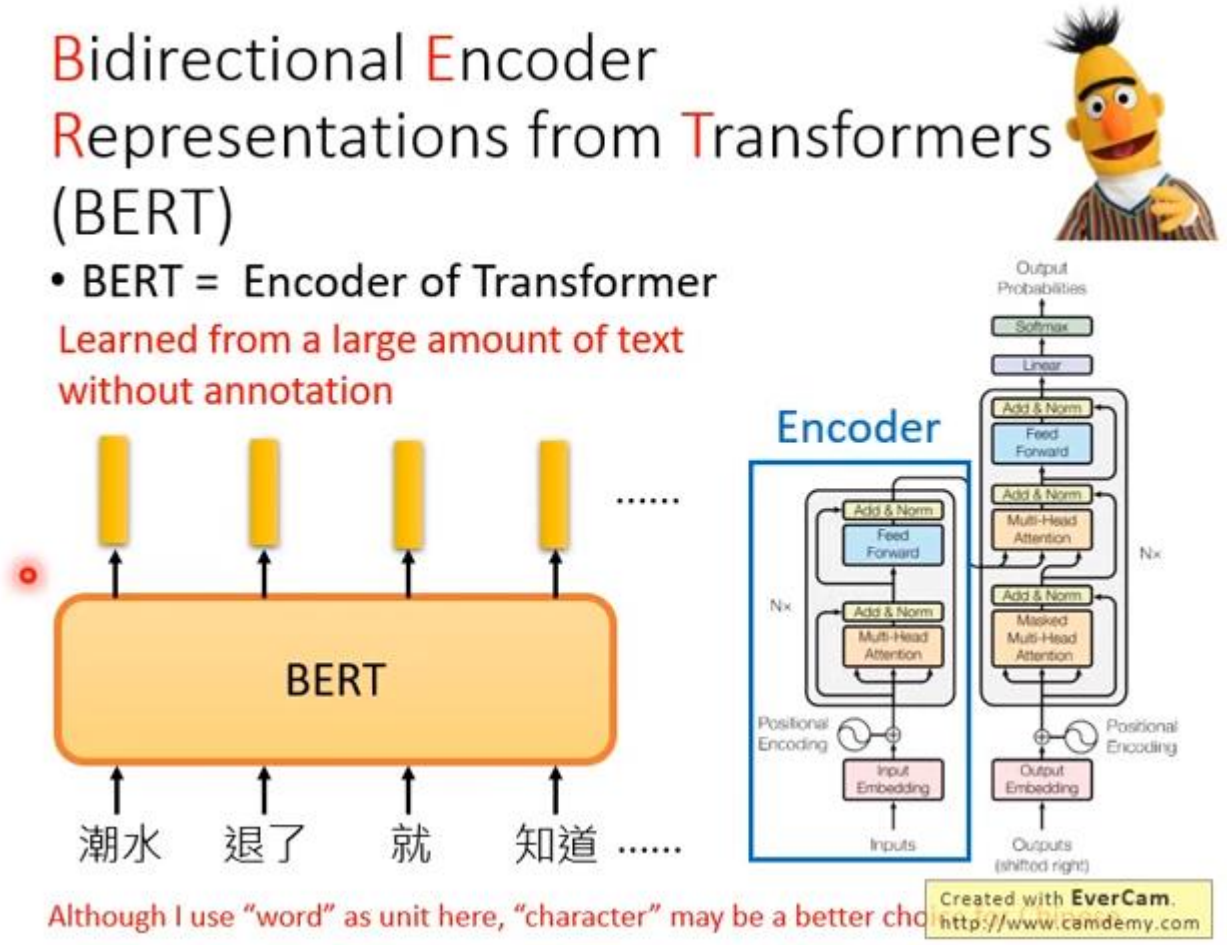
ELMo

- ELMo dynamically determines word embedding in downstream task.
- ELMo generates three embeddings:
  - word embedding
  - 1st LSTM layer embedding
  - 2st LSTM layer embedding

  2 hidden state embedding
- Pre-training -> get three embeddings (v_1, v_2, v_3) per word.
- Fine tuning -> freeze embeddings and train weights (w_1, w_2, w_3) for (v_1, v_2, v_3) per word.
- The final embedding is $w_1 v_1 + w_2 v_2 + w_3 v_3$

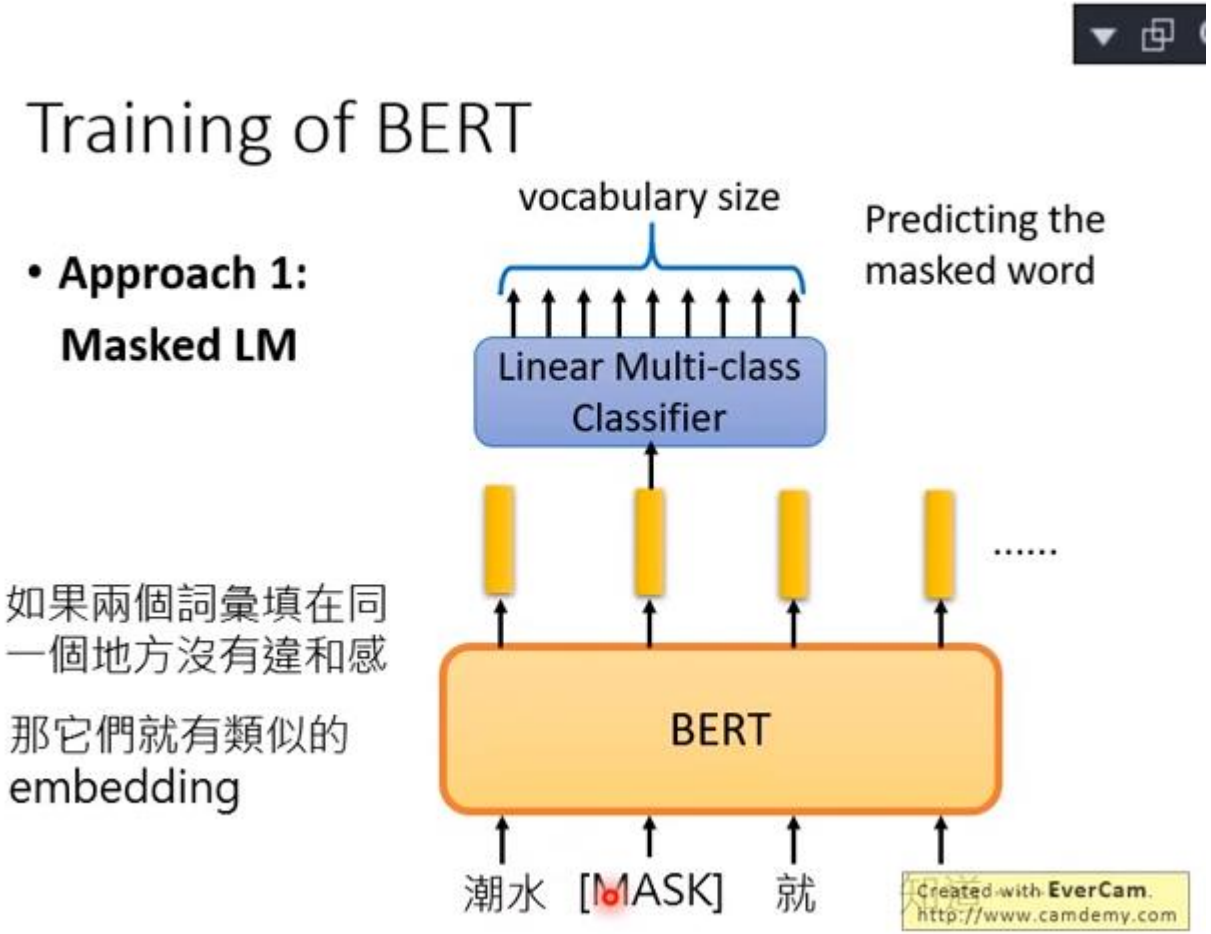
BERT

- **Transformer 的 Encoder**
- **輸出一串Embedding**



Approach 1 - Masked Language Model

- **預測被mask的詞彙**



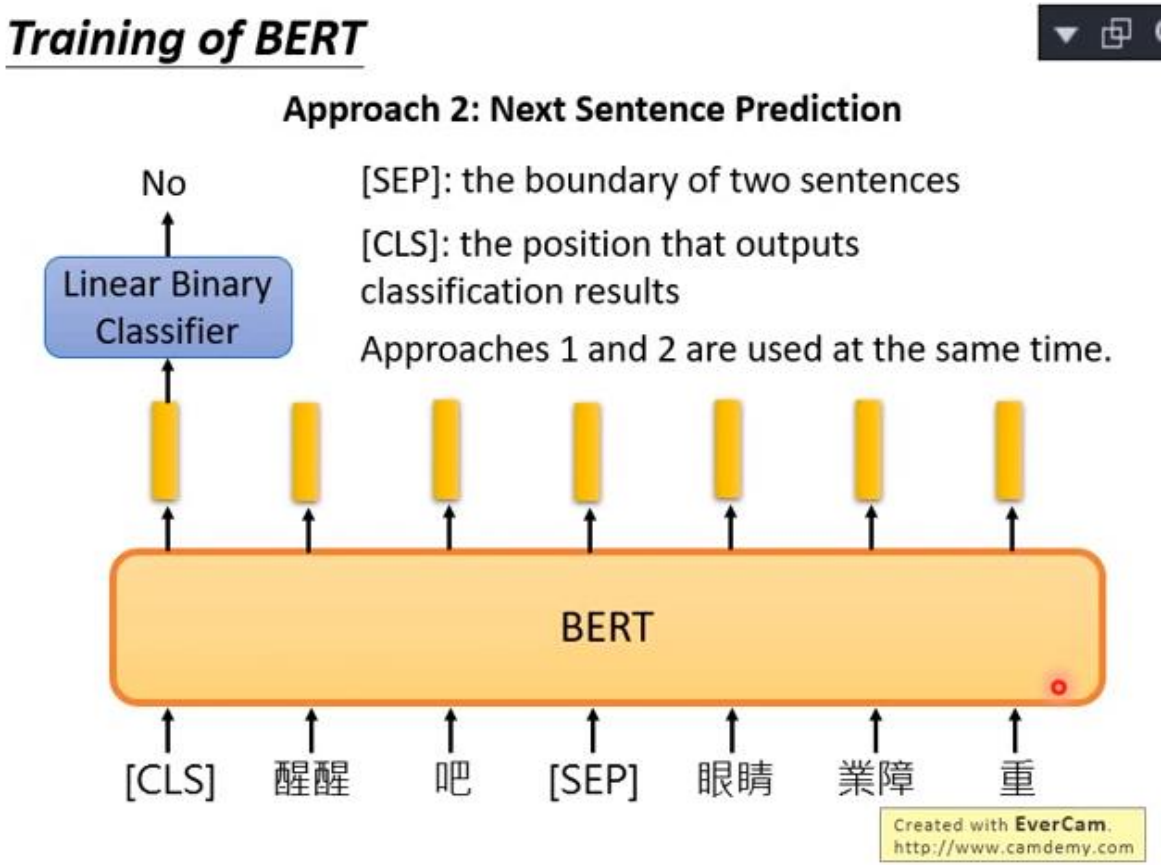
Approach 1 - Masked Language Model

- Estimate $p(x_i \mid x_{[1:i-1]}, x_{[i+1:n]})$ rather than $p(x_i \mid x_{[1:i-1]})$
  - Randomly mask 15% of all tokens and predict token
  - 80% of them - replace token with <mask>
  - 10% of them - replace with <random token>
  - 10% of them - replace with <token>

```
Alex is obnoxious but the tutorial   is awesome.
Alex is obnoxious but the <mask>     is awesome.
Alex is obnoxious but the <banana>   is awesome.
Alex is obnoxious but the <tutorial> is awesome.
```

Approach 2 - Next Sentence Prediction

- **兩種方法同時使用的效果最好**

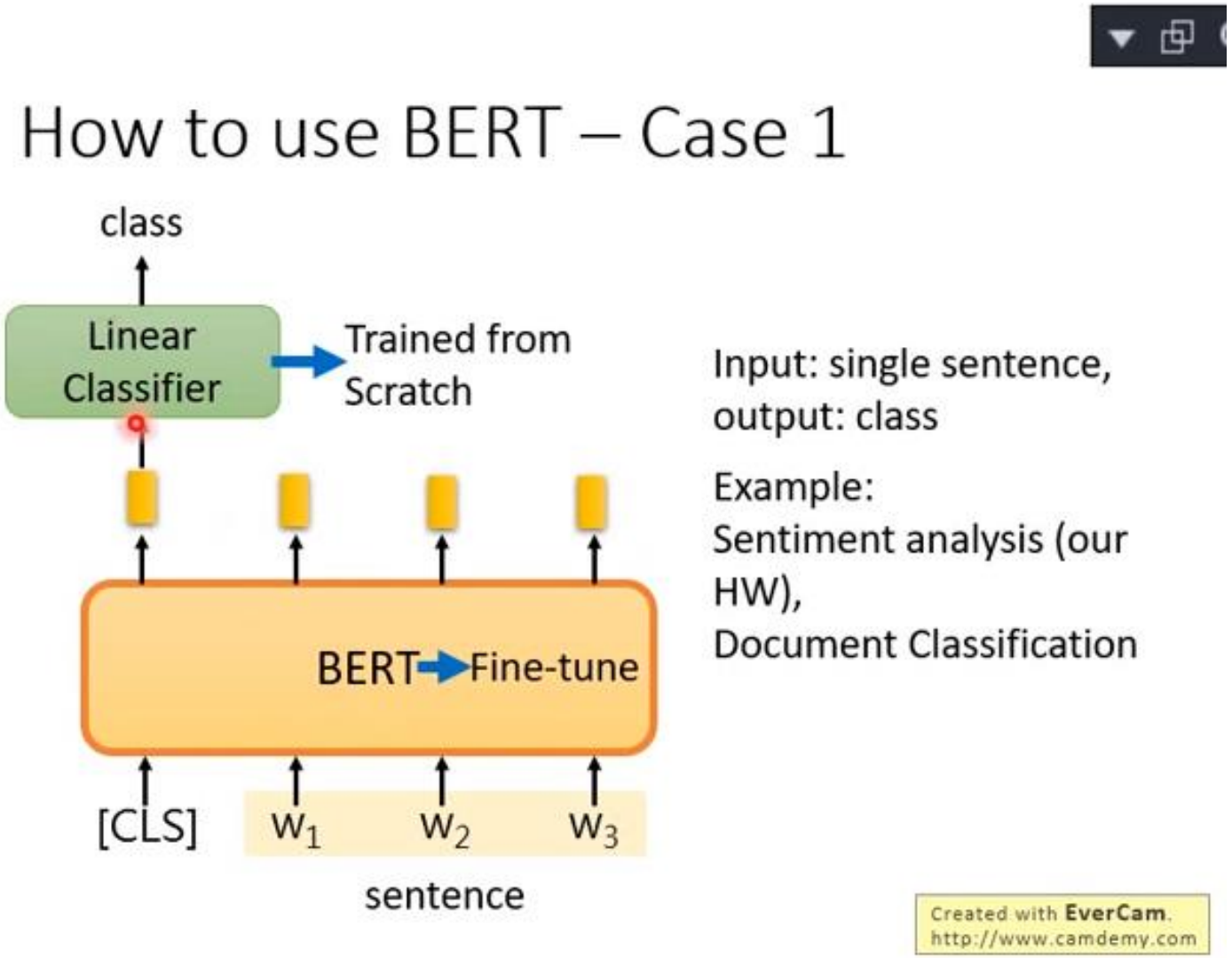


Approach 2 - Next Sentence Prediction

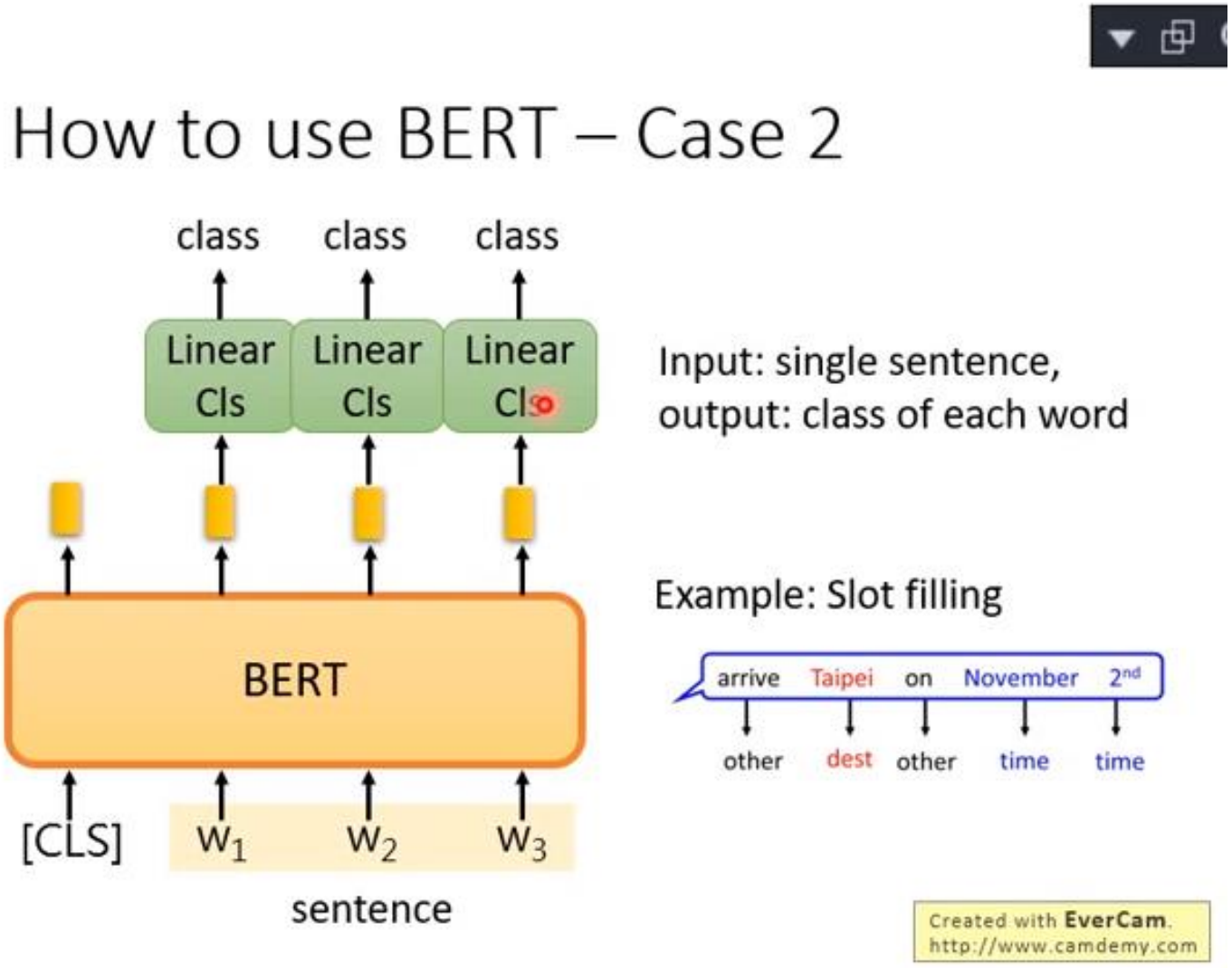
- Predict next sentence
  - 50% of the time, replace it by random sentence
  - Feed the Transformer output into a dense layer to predict if it is a sequential pair.
- **Learn logical coherence** 可以知道下面的句子對不對

<BOS> Alex is obnoxious <SEP> I don't like his shirt
<BOS> Alex is obnoxious <SEP> Look a Martian

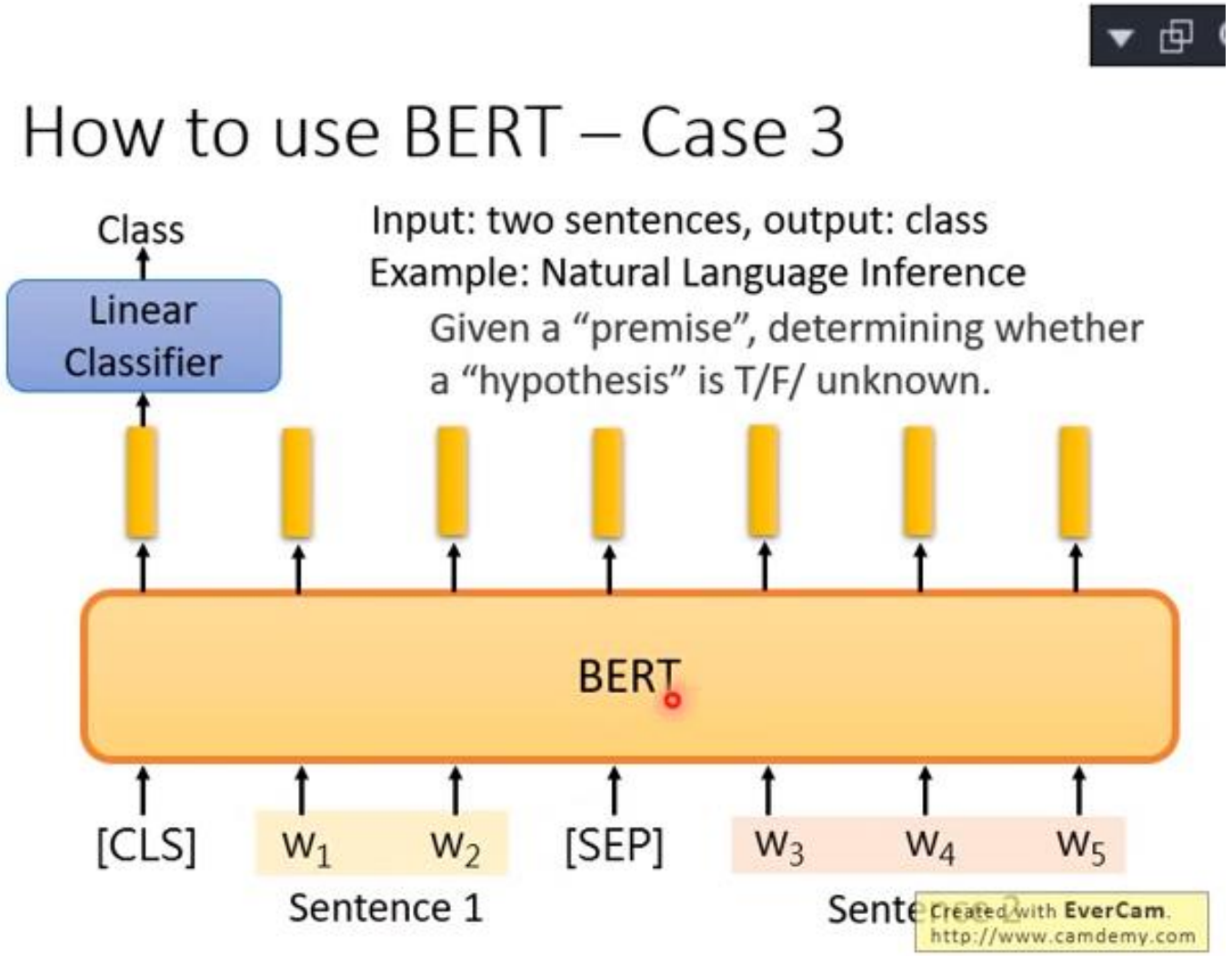
How to use BERT



How to use BERT



How to use BERT



How to use BERT

How to use BERT – Case 4

- Extraction-based Question Answering (QA) (E.g. SQuAD)

Document: $D = \{d_1, d_2, \cdots, d_N\}$

Query: $Q = \{q_1, q_2, \cdots, q_M\}$

output: two integers (s, e)

Answer: $A = \{d_s, \cdots, d_e\}$

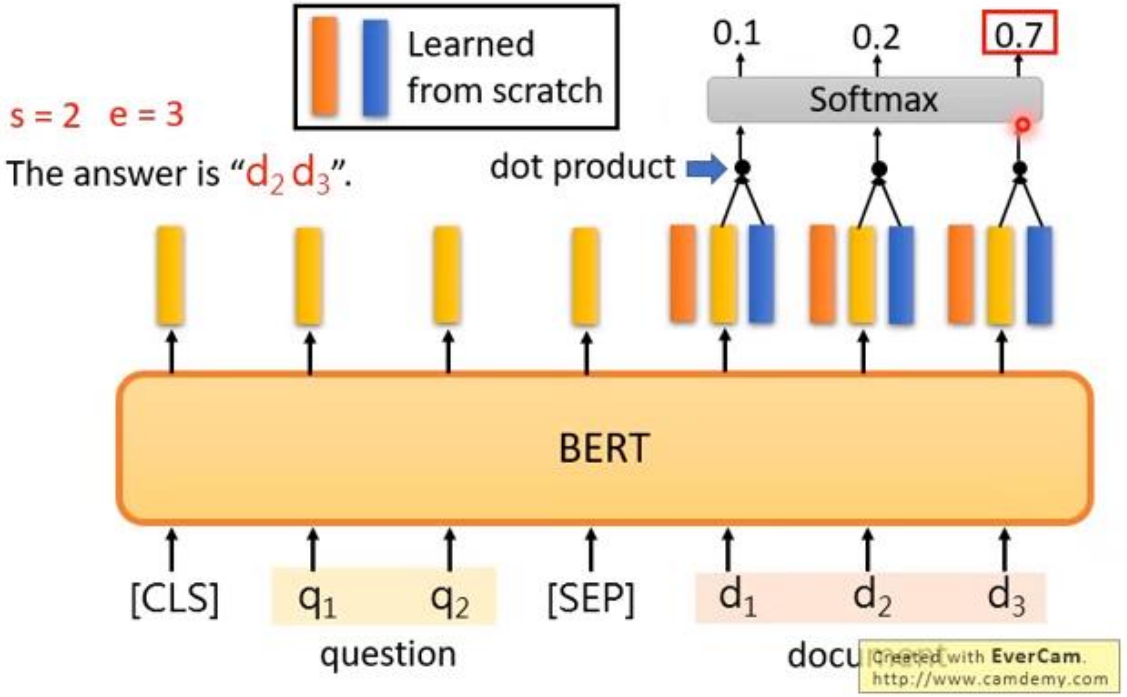
In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain 77 atte 79 cations are called "showers".

What causes precipitation to fall?
gravity $s = 17, e = 17$

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud $s =$

How to use BERT – Case 4



GPT

GPT uses Transformer Decoder (Radford et al., ‘18)

- Pre-train language model, then fine-tune on each task
- **Trained on full length documents**
- 12 blocks, 768 hidden units, 12 heads
- SOTA for 9 NLP tasks

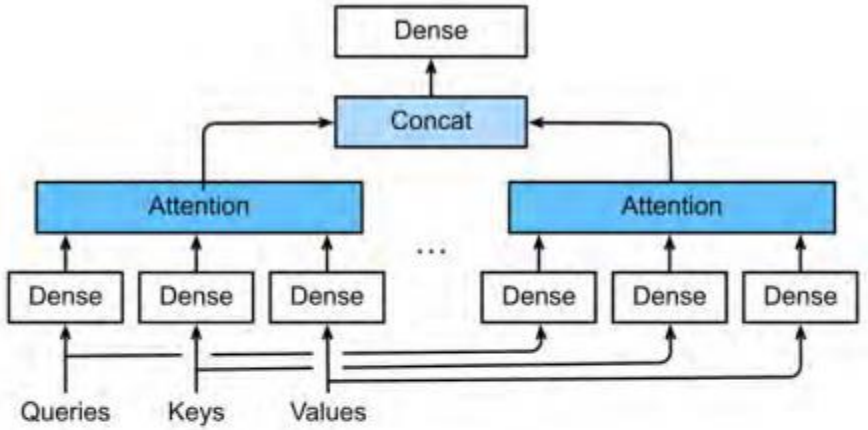
- Language model only looks **forward**
  - I went to the bank to deposit some money.
  - I went to the bank to sit down.

GPT

- **GPT-2的參數量遠大於GPT (至少10倍)**
- **Transformer 的 Decoder**
- **40GB 網際網路文本**

Architecture

- (Big) transformer encoder
- Train on large corpus (books, wikipedia) with > 3B words

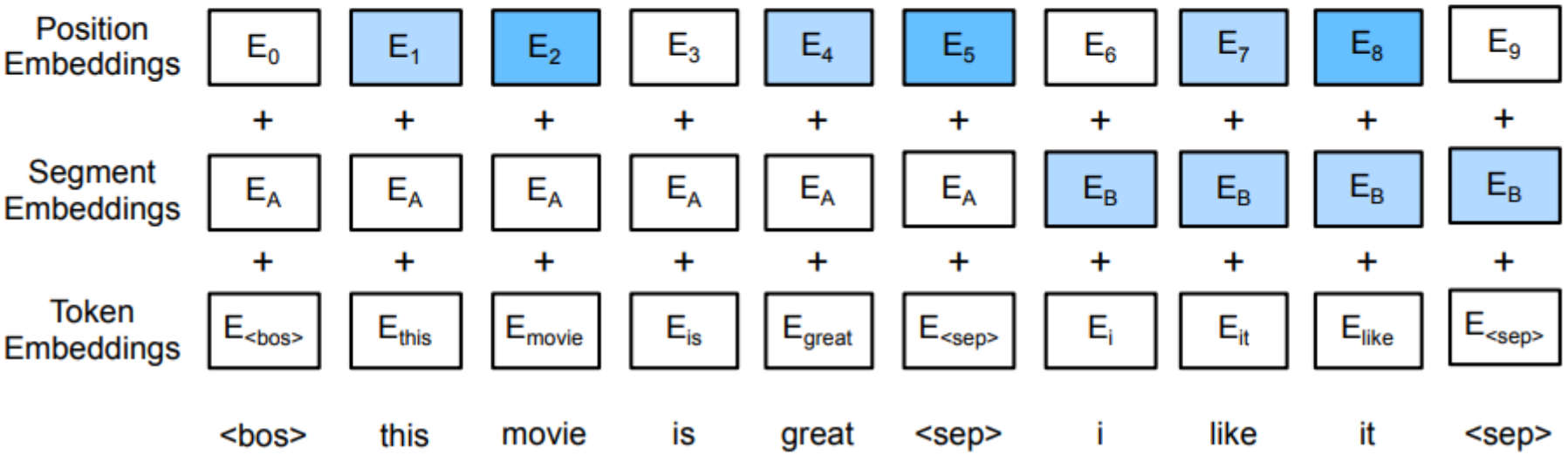


	blocks	hidden units	heads	parameters
small	12	768	12	110M
large	24	1024	16	340M

courses.d2l.ai/berkeley-stat-157/index.html

Input Encoding

- Each example is a pair of sentences
- Add segment embedding and position embedding



GPT2 (it gets even bigger, Radford et al., ‘19)

- Pretrained on 8M webpages (WebText, 40GB)
- Without fine-tuning SOTA on 7 language models

	blocks	hidden units	parameters
small	12	768	110M
large	24	1024	340M
GPT2	48	1600	1.5B

GPT2 Demo (gluon-nlp.mxnet.io)

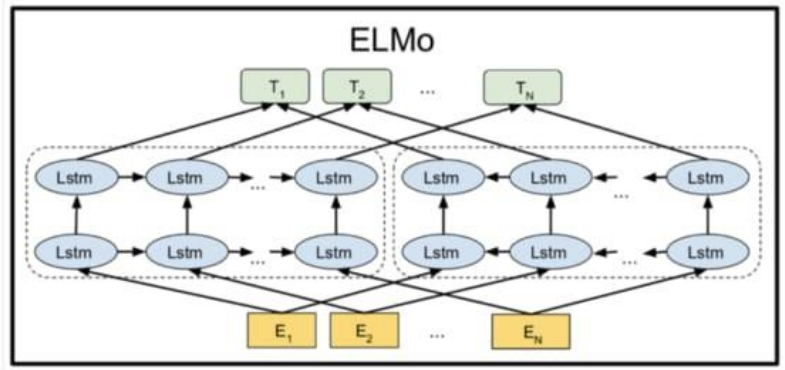
```
$python sampling_demo.py --model 117M
Please type in the start of the sentence
>>> average human attention span is even shorter than that of a
goldfish
-----  Begin Sample 0 -----
average human attention span is even shorter than that of a
goldfish strutting its way down the jaws. An estimate by the USA
TODAY Science team of 80 human-sized models reveals that a complex
jaw becomes a grandiose mitesaur in 100 million years, less than an
exothermic Holocene huge sea lion, and towering 500 meters tall.

Similar mitesaur-sized jaws would burden as trillions

Scientists would expect a lost at least four million times as much
time in the same distances ocean as other mammals
```

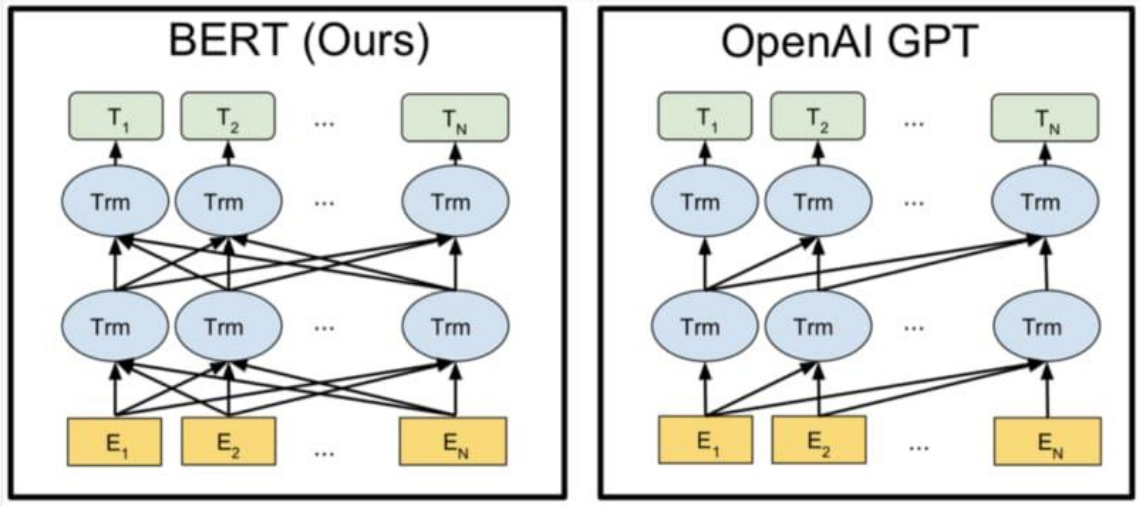

Q & A

- **Transformer, ELMO, GPT, BERT的目的 & 結構?**

三者間的關係

- ELMO: 動態Embedding
- GPT: 簡單使用Transformer的Decoder
- BERT: 使用Transformer的Encoder
  與克漏字來訓練





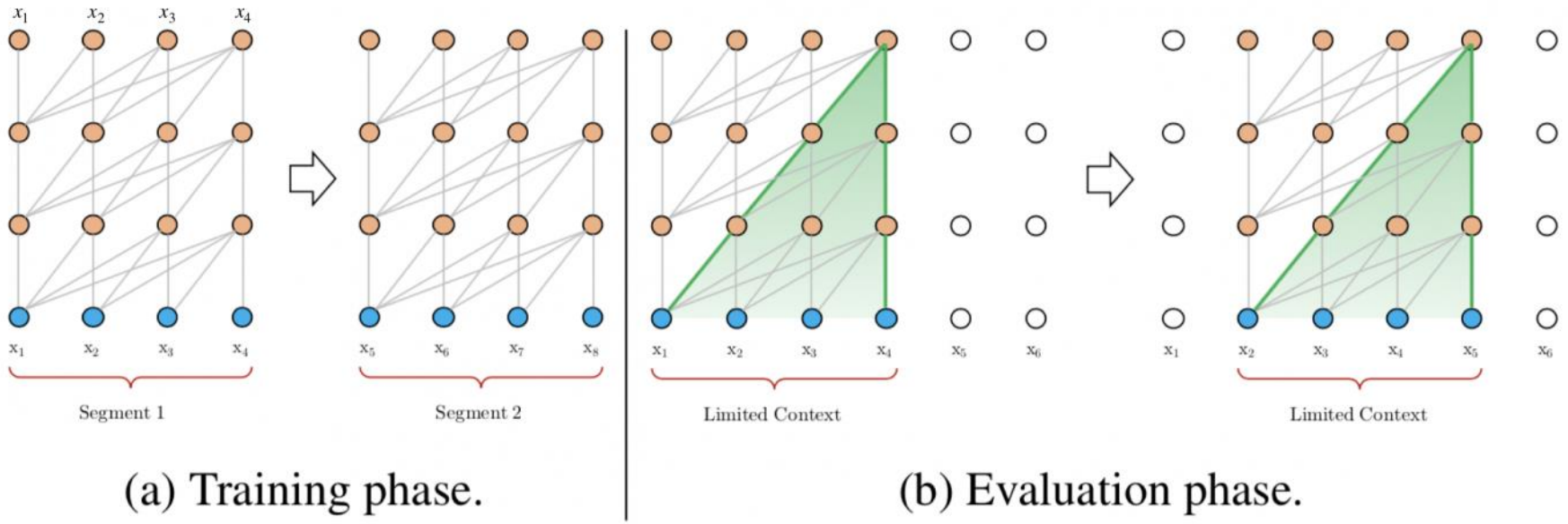
Challenges

And More

- Character-Level Language Modeling with Deeper Self-Attention (Rami Al-Rfou et al, 18’)
- Generating Long Sequences with Sparse Transformers (Rewon Child 1 et al, 19’)
- Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (Zihang Dai et al, 19’)
- Lightweight and Efficient Neural Natural Language Processing with Quaternion Networks(Yi Tay et al, 19’)
- XLNet: Generalized Autoregressive Pretraining for Language Understanding(Zhilin Yang et al, 19’)

Character-Level Language Modeling

- 添加多個loss來提高其表現以及加快擬合速度, 並加深transformer的層數
- character-level: 將句子按character為單位組成多個batch，每個batch預測一個詞
- 使用x1 … xn-1預測字符xn



https://www.infoq.cn/article/wt-KaTfcsAv9E7exzIkF

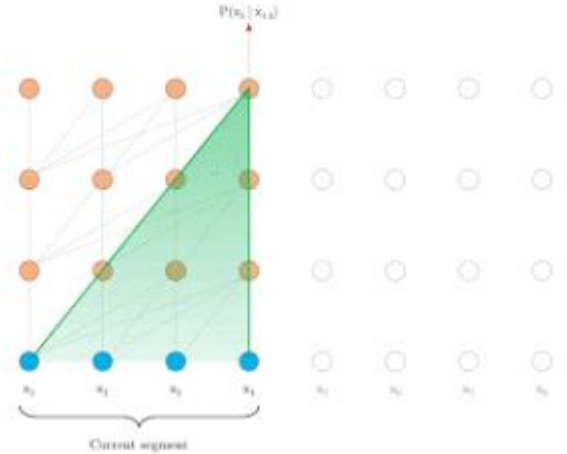
Transformer-XL

- 想要解決的問題：如何賦予Encoder捕獲長距離依賴的能力
- Transformer的序列長度極限在200左右, BERT的則在512
- 基於Al-Rfou等人提出的vanilla Transformer，並加入了
  - 循環機制（Recurrence Mechanism）
  - 相對位置編碼（Relative Positional Encoding）
- 可以被用於單詞級和character級的語言建模

https://zhuanlan.zhihu.com/p/70745925

Vanilla Transformer

- 每個segment分別訓練，之間不產生任何互動
- 隨t時刻移動預測character

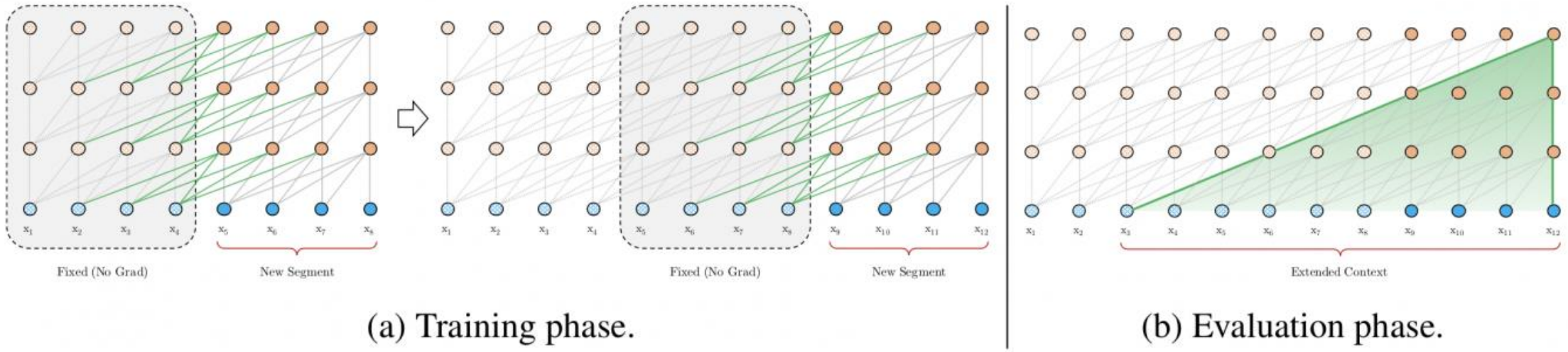


https://zhuanlan.zhihu.com/p/70745925

Transformer-XL

- 循環機制（Recurrence Mechanism）：
  - 利用之前的信息來實現長期依賴性
  - 灰色箭頭(同vanilla Transformer)為前一個hidden state的output
  - 綠色箭頭使用上一個segment的hidden state

跨segment



(a) Training phase.

(b) Evaluation phase.

https://www.infoq.cn/article/wt-KaTfcsAv9E7exzIkF

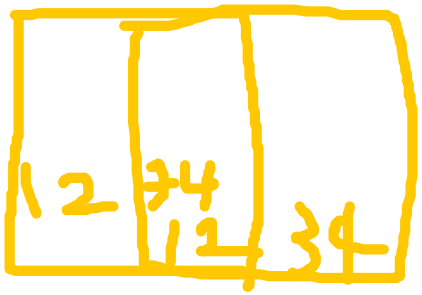
Transformer-XL

- 相對位置編碼（Relative Positional Encoding）
  - 原始Positional Encoding在每個segment分開處理，因此不同segment會有相同的Positional Encoding
  - 但他們的位置和重要性應該並不相同
  - 新的Encoding方式是根據相對距離

    W, u, v是要學習的參數

    E為片段s的詞向量

    R為相對位置



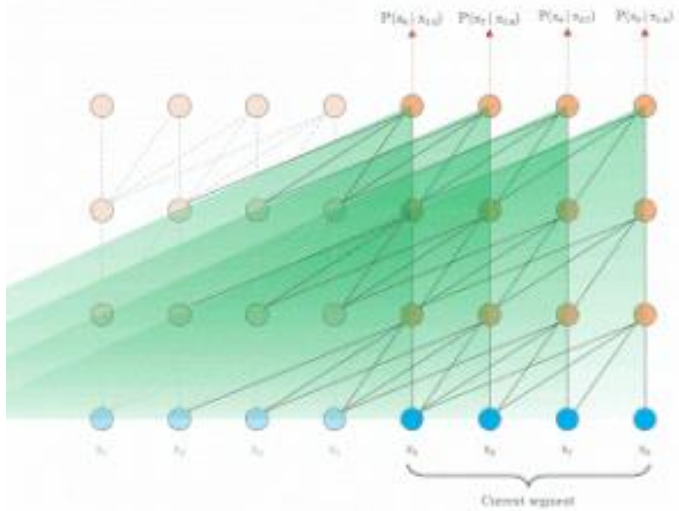
$$\mathbf{A}^{\mathrm{rl}}_{i,j} = \underbrace{\mathbf{E}^{\top}_{x_i}\mathbf{W}^{\top}_{q}\mathbf{W}_{k,E}\mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}^{\top}_{x_i}\mathbf{W}^{\top}_{q}\mathbf{W}_{k,R}\mathbf{R}_{i-j}}_{(b)} + \underbrace{u^{\top}\mathbf{W}_{k,E}\mathbf{E}_{x_j}}_{(c)} + \underbrace{v^{\top}\mathbf{W}_{k,R}\mathbf{R}_{i-j}}_{(d)}$$

https://www.infoq.cn/article/wt-KaTfcsAv9E7exzIkF

Transformer-XL

- 在不需要重新計算的情況下處理新段中的所有元素，顯著提高了速度
  (每個input都做預測，vanilla只在最後做預測)

input sequence=true



https://www.infoq.cn/article/wt-KaTfcsAv9E7exzIkF

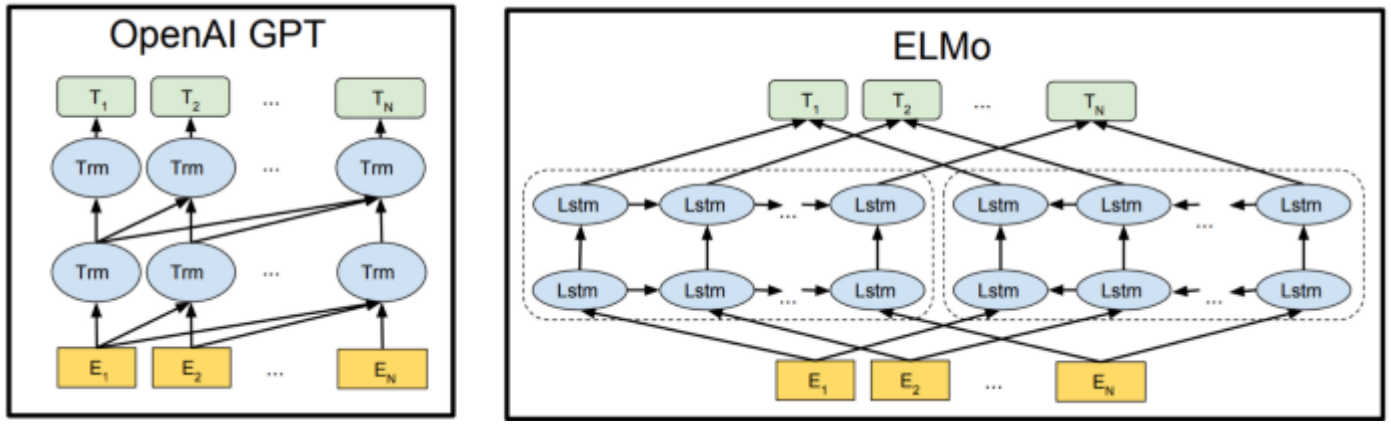
XLNet

- 改善了ELMo, GPT, BERT的缺點
- 有ELMo, GPT的AR (AutoRegressive)性質
- 也有BERT的AE性質，能夠捕捉bidirectional context的訊息
- 最後再使用Transformer-XL訓練大型文本的架構
- NLP趨勢: Pre-train model + Downstream(transfer learning)

https://medium.com/ai-academy-taiwan/2019-nlp%E6%9C%80%E5%BC%B7%E6%A8%A1%E5%9E%8B-xlnet-ac728b400de3

XLNet

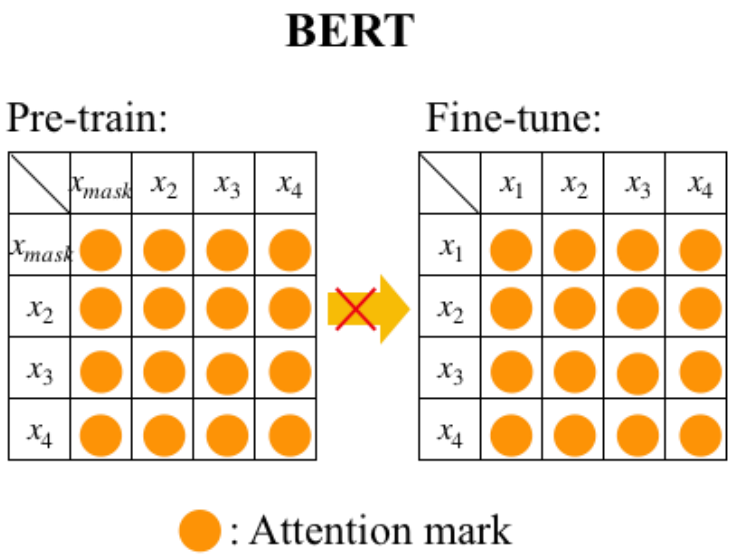
- 自回歸語言模型（AutoRegressive LM）
  - 從左向右的語言模型任務
  - AR的思想就是一個詞只能從上文或者下文任一個方向來判斷
  - ELMo講簡單一點就是一次訓練兩個AR，本質上還是一個AR模型。



https://zhuanlan.zhihu.com/p/70257427

XLNet

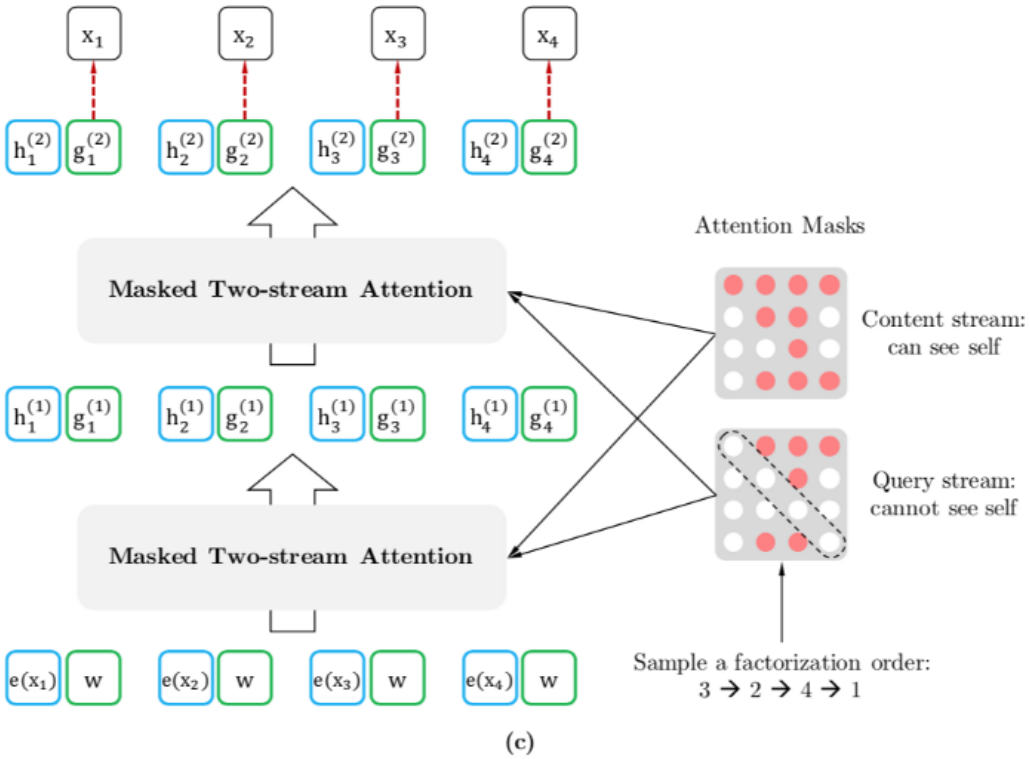
- 自編碼語言模型（AutoEncoding LM）
  - BERT的MLM(Masked Language Model)
  - 能夠抓到上下文的資訊
  - Input noise: 只有Pre-train會用到<mask>, Finetune時則不會, 且mask後會失去部分位置資訊



https://zhuanlan.zhihu.com/p/70257427

XLNet

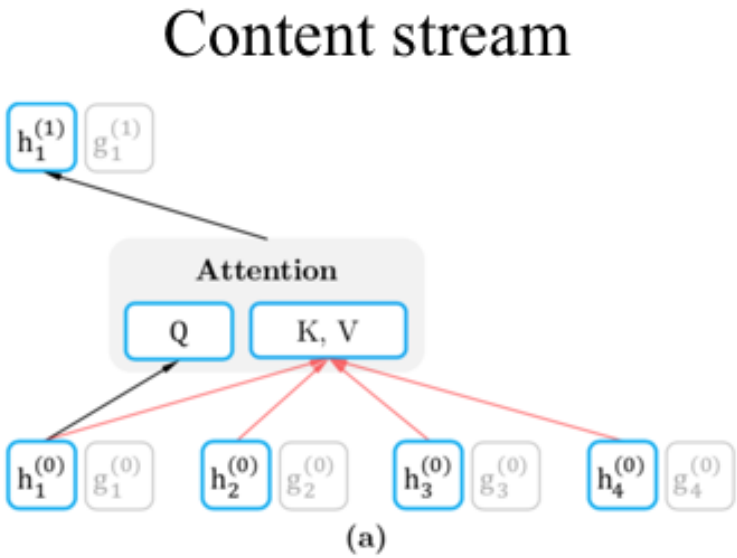
- Two-Stream Self-Attention
  - 使用Content stream以及Query stream
  - Content stream負責學習上下文
  - Query stream取代<Mask> token，把Content stream產生的representation拿來做預測



XLNet

- Content stream
  - 就是一個標準的self-attention
  - h1在進行Attention時會使用到QKV
  - h1作為Q，h1~4作為K與V

    (h1與h1~4進行attention)
  - 接著再與V相乘得到h1的representation

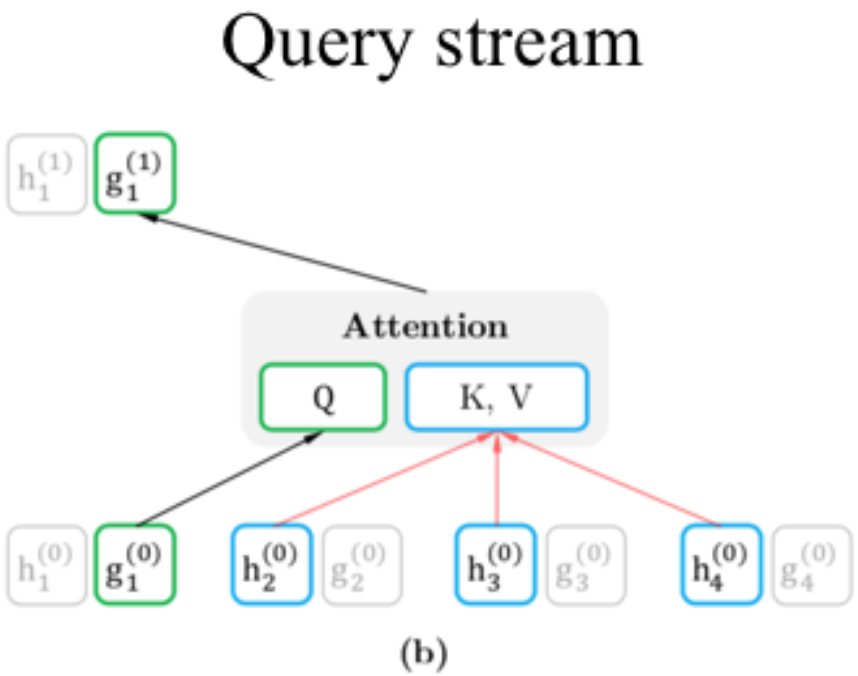


https://medium.com/ai-academy-taiwan/2019-nlp%E6%9C%80%E5%BC%B7%E6%A8%A1%E5%9E%8B-xlnet-ac728b400de3

XLNet

- Query Stream
  - 在Pre-train擔任預測單詞的作用
  - 另外設置一個representation g
  - 把當前t位置的attention weight mask掉

    (Attention Mask)



https://medium.com/ai-academy-taiwan/2019-nlp%E6%9C%80%E5%BC%B7%E6%A8%A1%E5%9E%8B-xlnet-ac728b400de3

謝謝聆聽

Thank you