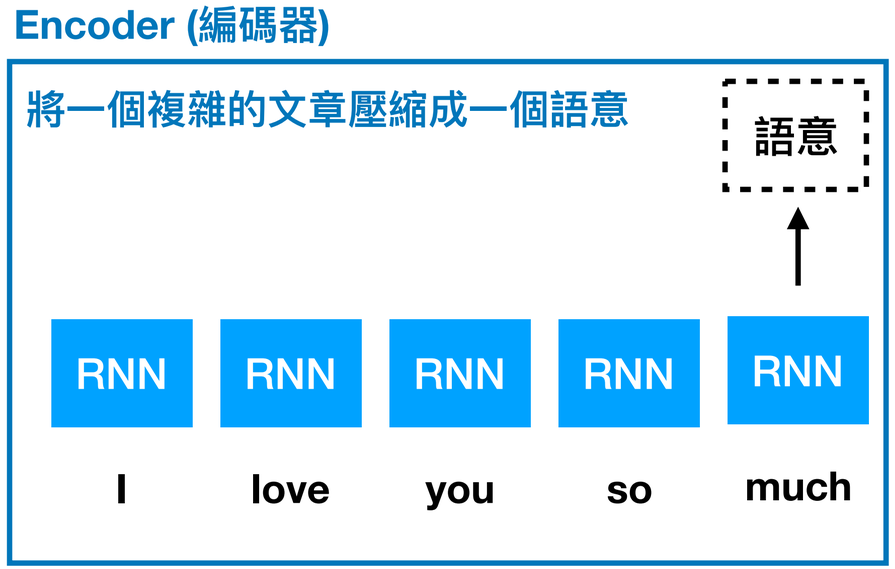
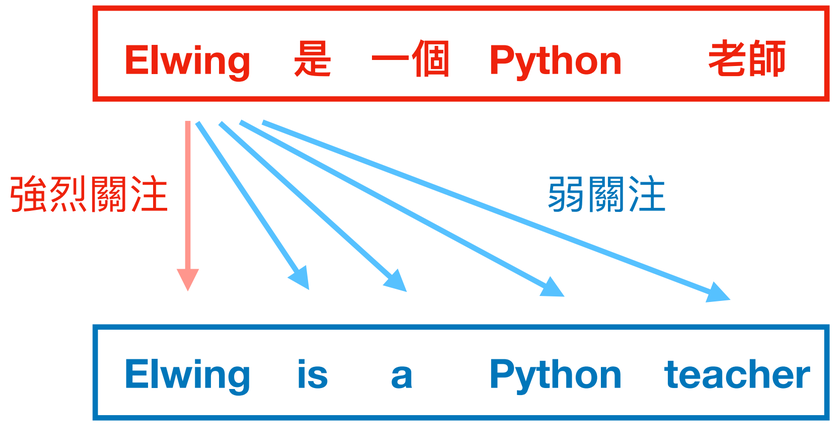
Seq2Seq問題

我們上一個章節看到了 Seq2Seq 可以幫我們把一個序列翻譯成另外一個序列，但是問題是 Encoder 是把整篇文章濃縮成一個語意，所以一些詞語的語意不免不夠精確或丟失了！



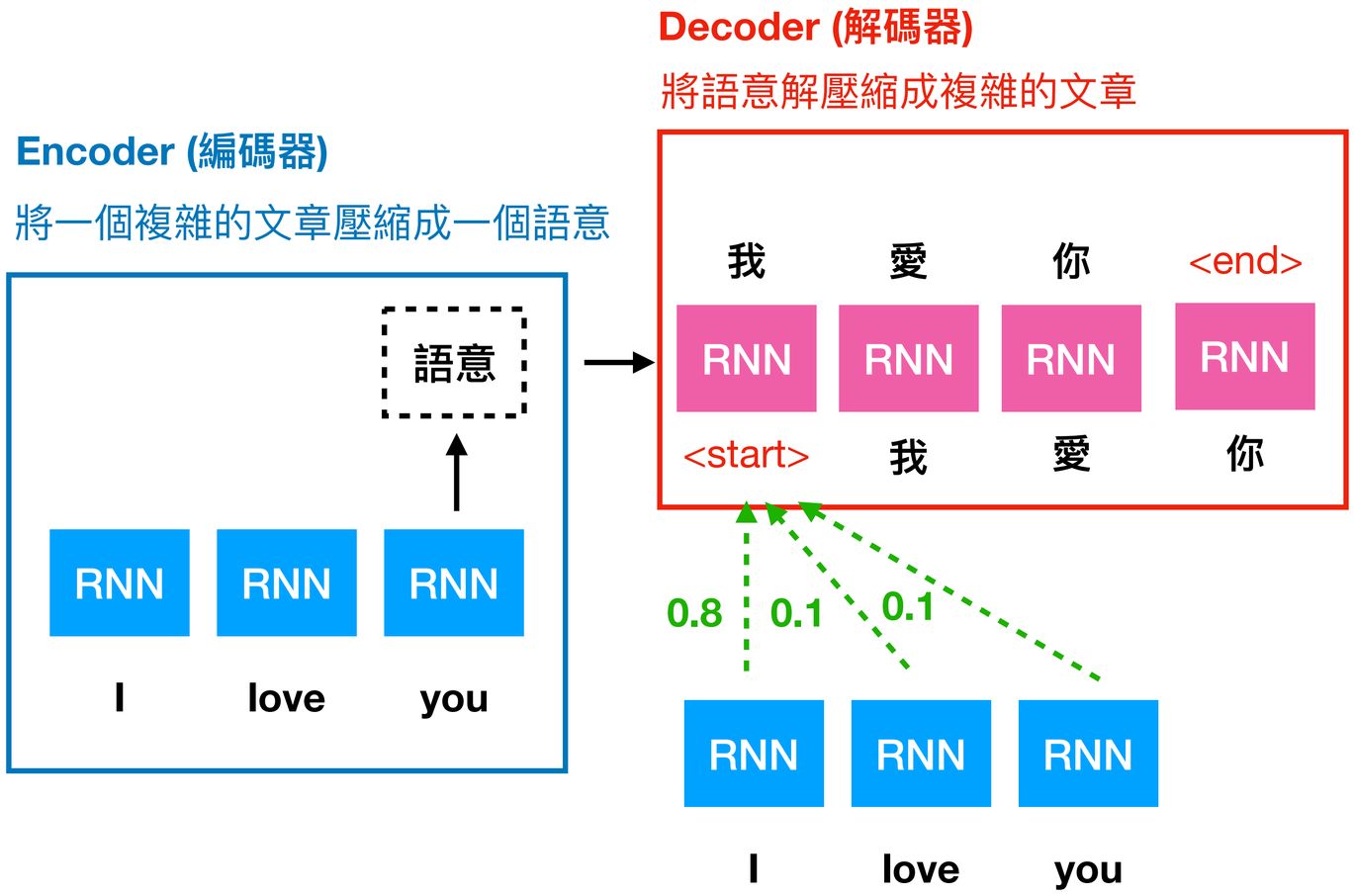
解決辦法

於是我們就想，我們有沒有一個好辦法，可以關注每一個詞，
就像人類真正在翻譯一樣，翻譯的時候『注意』原本的詞彙



Attention

於是 Attention 機制就來了，Attention機制除了關心『累積語意』以外還關心『來源單詞該得到的關注度』

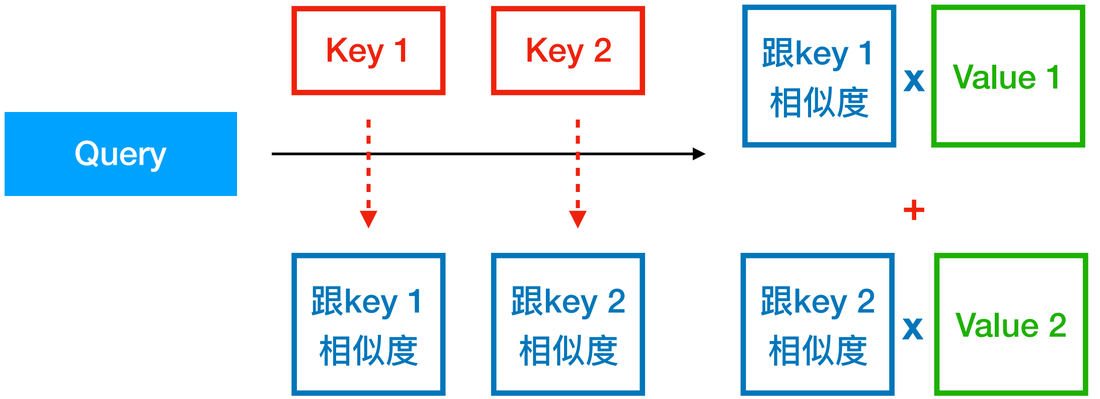


簡化一點

其實說的簡單一點，就是為你的每一個 input 的語意做出一個
『查詢向量』，讓我們可以查詢『輸出』和『輸入』的『相關性』

可以是跟 Value 一樣 或者是乘上係數 W x Value

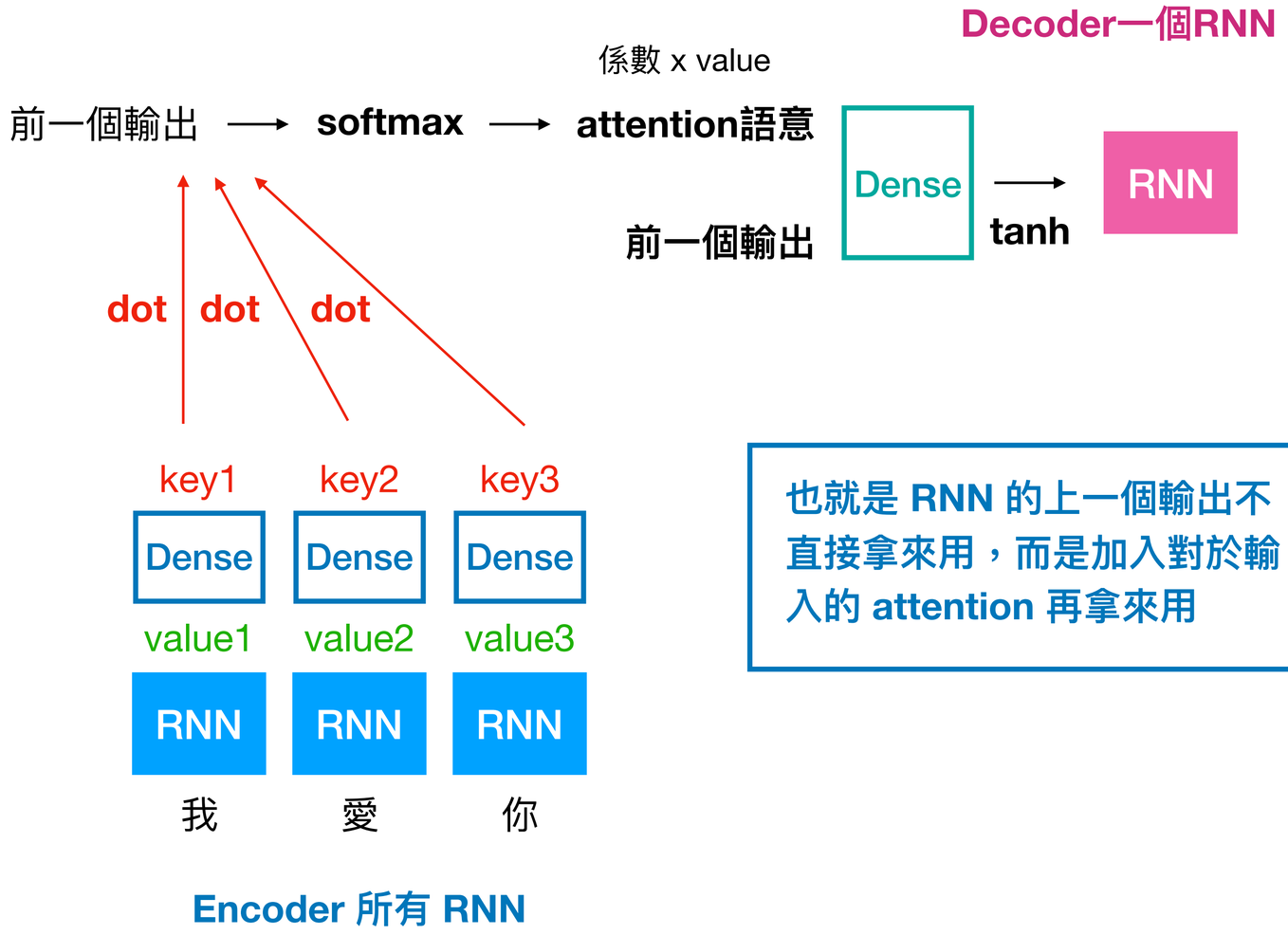
就是你input每個詞的Embedding



相似度可以取cos相似度

但要做一個softmax轉化，加起來等於1

Attention步驟

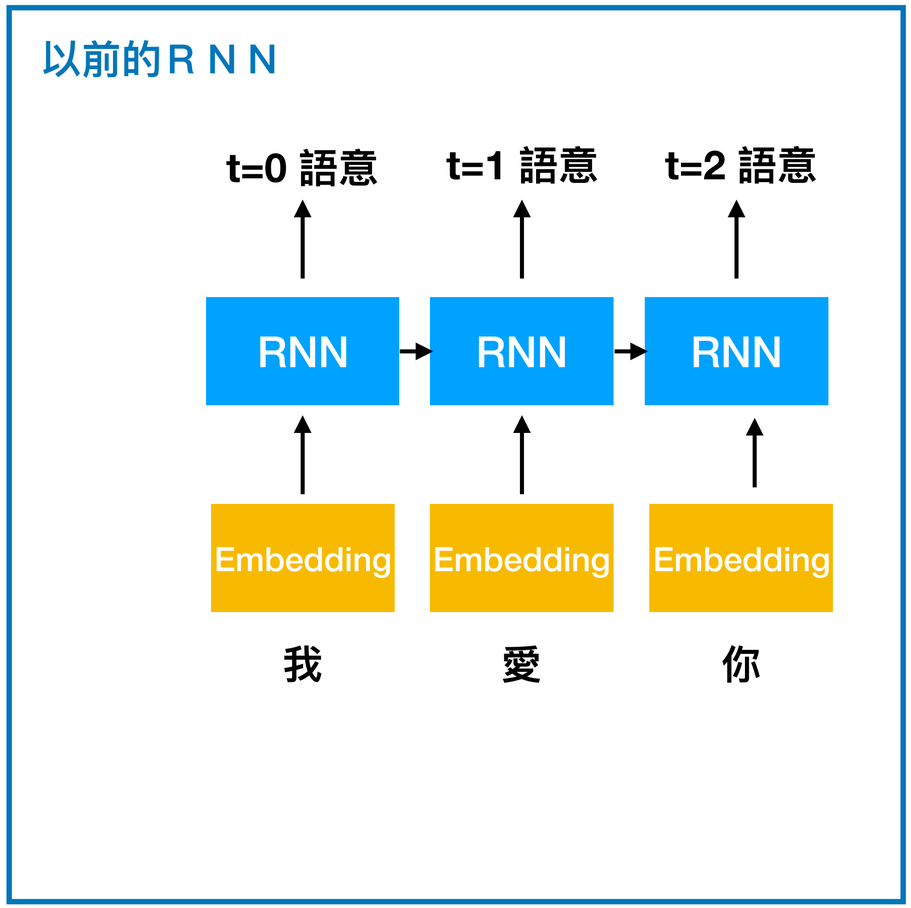


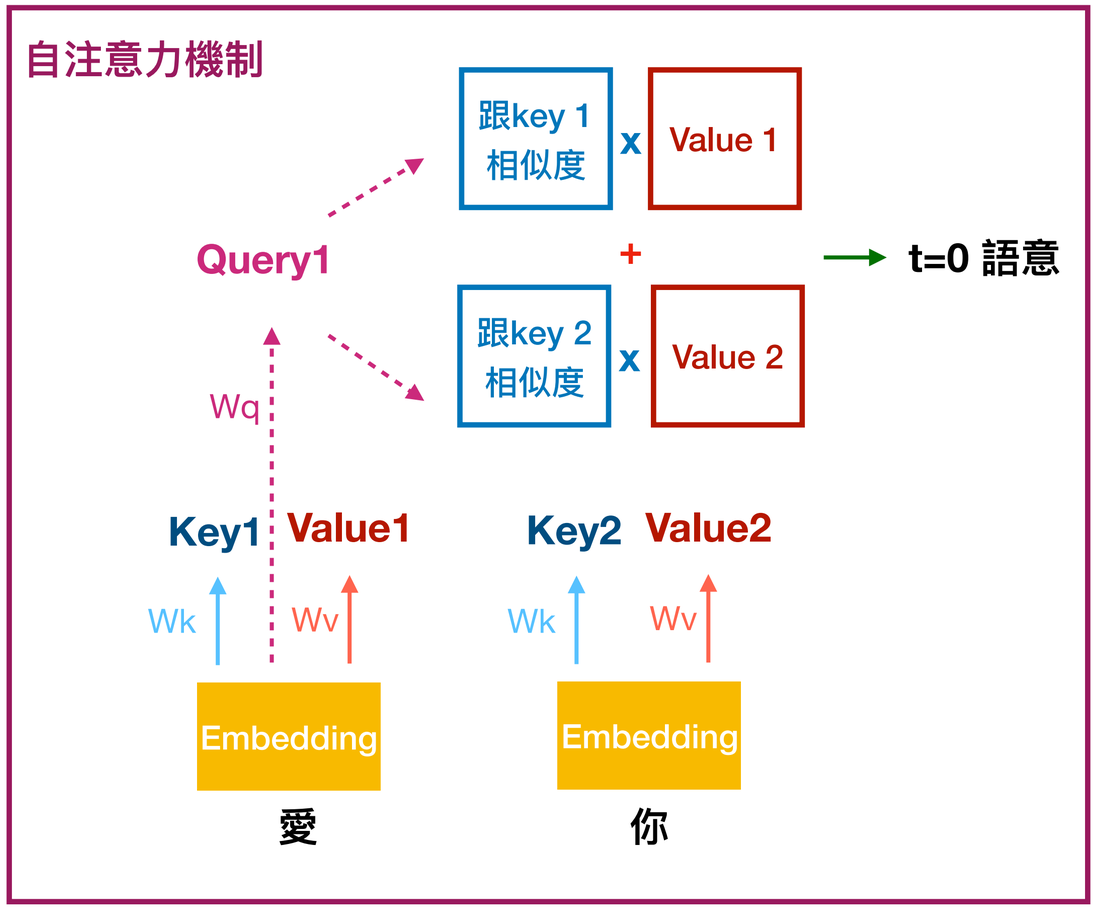
也就是 RNN 的上一個輸出不直接拿來用，而是加入對於輸入的 attention 再拿來用

Self Attention

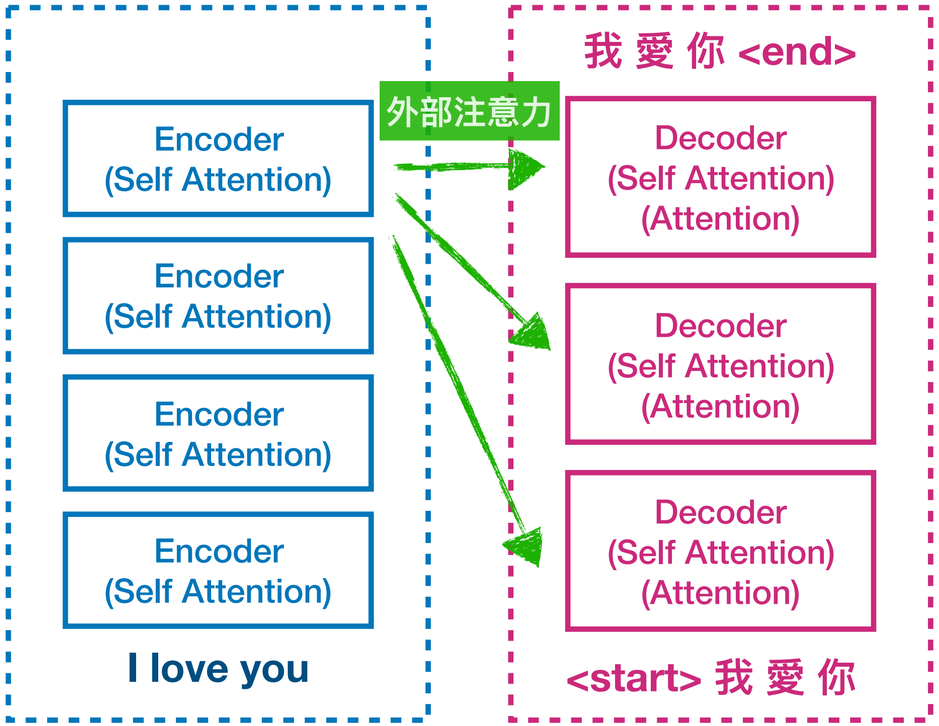
看完上面的章節，我們想到一件事，Attention太棒惹吧！我們可以『跨越時空』看到所有的對象詞彙，並且做出相似度的比較來得到『注意力向量』，那我就有個魔鬼的想法了！就是，我們可不可以連 Encoder 地方的 RNN 都拆掉，完全用 Attention機制來取代，這就是 Google 提出的一篇驚天動地的論文『Attention is all you need』，你不再需要 RNN, LSTM, GRU 等等的，你唯一需要的！就是 Attention，這種在模型的內部也直接使用的 Attention 我們就叫做 『Self Attention』『自注意力機制』

自注意力機制


以前的ＲＮＮ
t=0 語意
t=1 語意
t=2 語意
RNN
RNN
RNN
Embedding
Embedding
Embedding
我
愛
你

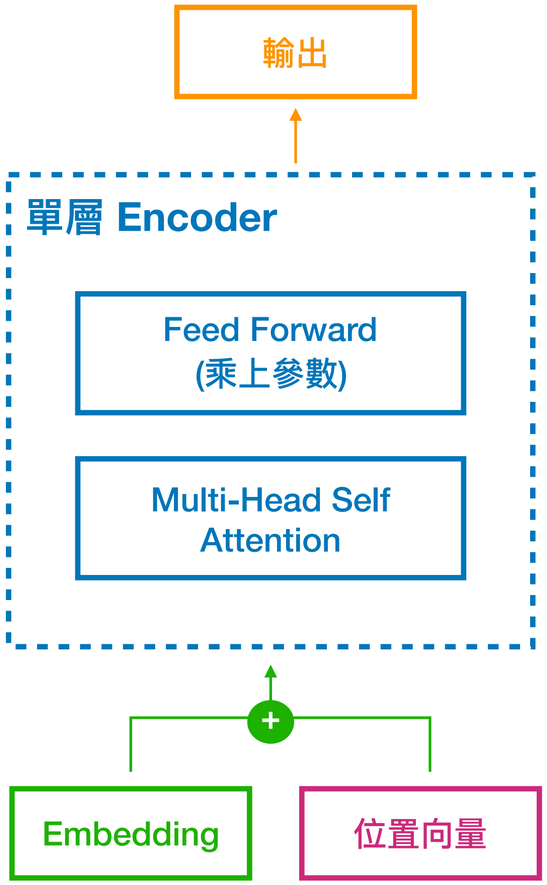

自注意力機制
跟key 1
相似度
x
Value 1
Query1
+
t=0 語意
跟key 2
相似度
x
Value 2
Wq
Key1
Value1
Key2
Value2
Wk
Wv
Wk
Wv
Embedding
Embedding
愛
你

Transformer



Google 在 『Attention is all you need』提出了 Seq2Seq 的注意力版本，完全用 Attention 取代了 RNN/LSTM/GRU，我們一起來看看 Transformer 的架構

Encoder



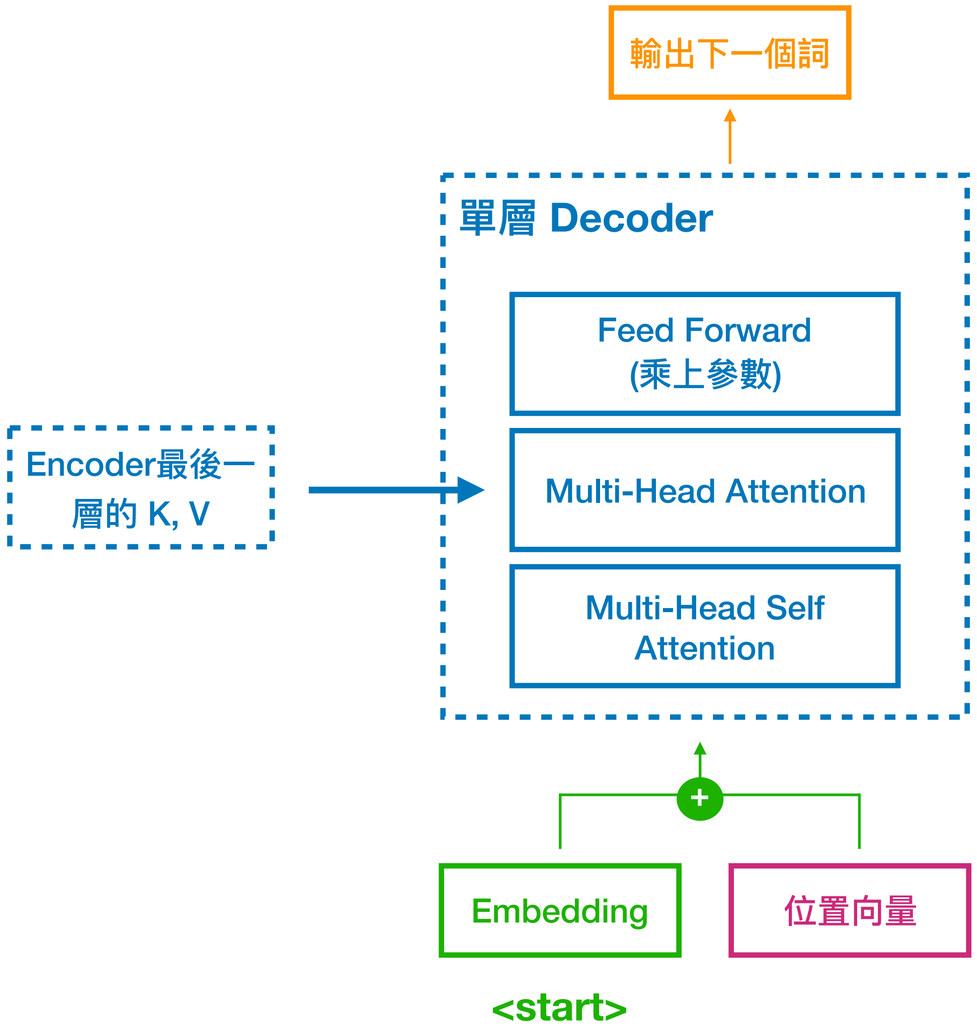
多頭 Self Attention

你不覺得一組Q, K, V就很像卷積網路的萃取特徵嗎？我們當然可以有多組，譬如我們可以用不同的參數產生了8組的 Q, K, V，在把八組組合起來變成一個output

位置向量

其實 Attention 是沒有順序的，因為你只是對於輸入的所有詞得到衡量係數，那怎麼加上位置資訊呢？很簡單，把位置也當成一個輸入，得到位置Embedding，一樣參數讓模型自己學習

Decoder



差別

只差在 Decoder 多加上一層普通的 Attention Layer，根據輸入的字串做 Attention，輸入和輸出基本上跟 Seq2Seq一致

Self Attention & Attention

你會發現，其實 Self Attention 和 Attention 只是關注的對象不一樣而已，一個是網內互打(只關注自己)，另外一個是網外，所以我其實也喜歡叫做 Intra-Attention 和 Inter-Attention

BERT

BERT

終於，我們來到了赫赫有名的BERT，BERT全名是 Bidirectional Encoder Representation from Transformers，所以其實沒什麼新的東西！就是把上面介紹的 Transformer 裡面的 Encoder 部分拿出來而已！

目標

如果把 Decoder 當作你的『下游任務』，那 Encoder 不就是你的語意萃取嗎？ 我們前面也說到了，要訓練一個 NLP 模型需要的語料是非常非常龐大，所以 BERT 就是 Google的『佛心』！幫你訓練好語意萃取，你只要在 BERT 後面接上你想要的任務就可以了！