TEXT SUMMARIZE & BERTVIZ

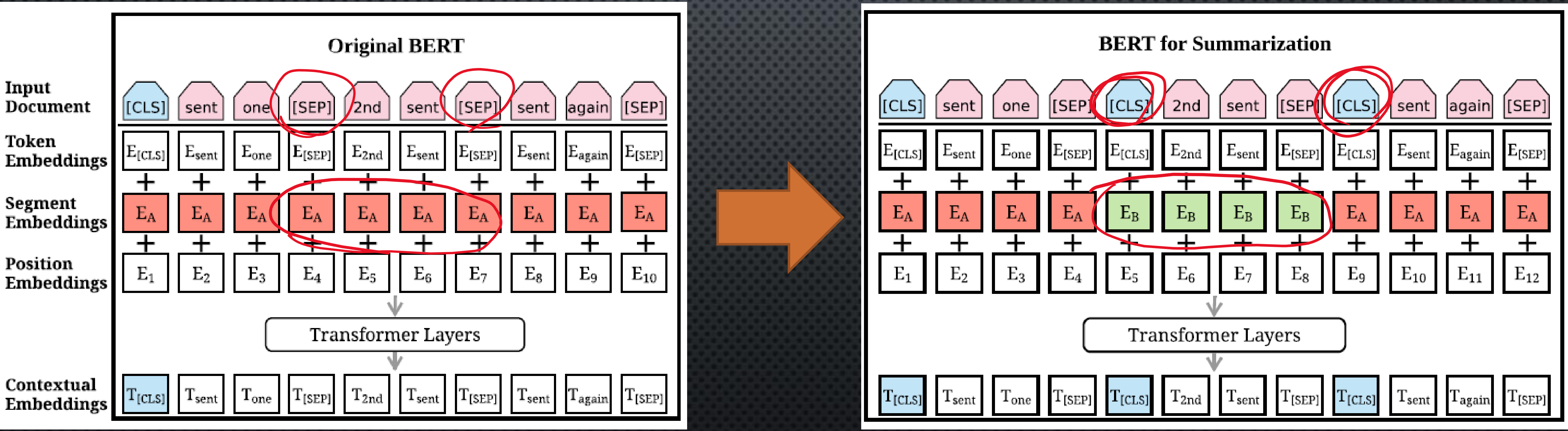
OUTLINE

- 使用Transformer擷取文章摘要
- 使用BertViz視覺化模型的Attentions

從Bert Finetune來處理摘要擷取(text summarization)

CLS: sentence embedding>用來後續的預測任務

E_A, E_B: segmentation embeddings>根據奇數偶數句來區分



BERT-SUM-TEXT

[1908.08345] TEXT SUMMARIZATION WITH PRETRAINED ENCODERS (ARXIV.ORG)

- 什麼是extractive summarization?
  - 對每個句子做Binary classification
    - 是重點/不是重點

簡單來說，既然前面已經對句子做embedding，
我們丟進去分類器的單位就不是token為單位，而是sentence

優點> 取出的摘要都是完整的句子
缺點> 沒有辦法重組句子(精煉)

Language Model : Unigram

- $p(w_1, w_2, w_3, w_4, w_5 \ldots w_n)$

$= P(w_1) \cdot P(w_2) P(w_3) \cdots P(w_n)$

- p(今天，是，春节，我们，都，休息) ✓

= P(今天)·P(是)·P(春天)·P(我们) P(都) P(休息)

- p(今天，春节，是，都，我们，休息)

= P(今天)·P(春天) P(是) P(都) P(我们) P(休息)

Unigram: 假設詞語之間是互相獨立 (不考慮順序)

Bigram: 詞語出現的機率和前一個詞有關
(考慮兩個詞的順序)

Language Model : Bigram ← 1st order markov assumption

- $p(w_1, w_2, w_3, w_4, w_5 \ldots w_n)$

$= P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2) \cdots P(w_n|w_{n-1}) = P(w_1) \cdot \prod_{i=2}^{n} P(w_i|w_{i-1})$

- p(今天，是，春节，我们，都，休息)

= P(今天)·P(是|今天)·P(春节|是)·P(我们|春节)·P(都|我们)·P(休息|都)

- p(今天，春节，是，都，我们，休息)

= P(今天)·P(春节|今天)·P(是|春节)·P(都|是)·P(我们|都)·P(休息|我们)

R1: Rouge 1
R2: Rouge 2
RL: RougeL

$$ROUGE-N = \frac{\sum_{S \in References} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in References} \sum_{gram_n \in S} Count(gram_n)}$$

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

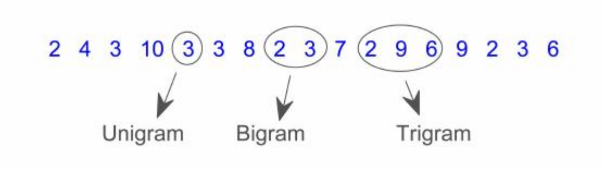
$$ROUGE-L = F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

句子出現的合理性(機率)

字詞出現的機率

Unigram: $P(w_1, w_2, \cdots, w_m) = \prod_{i=1}^{m} P(w_i)$

Bigram: $P(w_1, w_2, \cdots, w_m) = \prod_{i=1}^{m} P(w_i | w_{i-1})$



+ R1 : police killed the gunman.
+ R2 : the gunman was shot down by police.
- C1 : police ended the gunman.
- C2 : the gunman murdered police.

Scoring	相對於誰	Calc	
R1	C1>R1,R2	(3+3)/(4+7)=6/11	C1和R1,R2個重複了3個詞 R1,R2分別有4,7個詞
R1	C2>R1,R2	(3+3)/(4+7)=6/11	
R2	C1>R1,R2	(1+1)/(3+6)=2/9	重疊的bigram只有1個(the gunman) R1,R2分別有3,6個bigram
R2	C2>R1,R2	(1+1)/(3+6)=2/9	
RL	C1>R1	Avg(3/4, 3/4)=3/8	LCS=police the gunman(3) R1,C1長度=4

Model	R1	R2	RL
Oracle	52.59	31.24	48.87
Lead-3	40.42	17.62	36.67
Extractive			
SummaRuNNer (Nallapati et al., 2017)	39.60	16.20	35.30
Refresh (Narayan et al., 2018b)	40.00	18.20	36.60
Latent (Zhang et al., 2018)	41.05	18.77	37.54
NeuSum (Zhou et al., 2018)	41.59	19.01	37.98
Sumo (Liu et al., 2019)	41.00	18.40	37.20
TransformerExt	40.90	18.02	37.17
Abstractive			
PTGen (See et al., 2017)	36.44	15.66	33.42
PTGen+Cov (See et al., 2017)	39.53	17.28	36.38
DRM (Paulus et al., 2018)	39.87	15.82	36.90
BottomUp (Gehrmann et al., 2018)	41.22	18.68	38.34
DCA (Celikyilmaz et al., 2018)	41.69	19.47	37.92
TransformerAbs	40.21	17.76	37.09
Bert-based			
BertSumExt	43.25	20.24	39.63
BertSumExt w/o interval embeddings	43.20	20.22	39.59
BertSumExt (large)	43.85	20.34	39.90
BertSumAbs	41.72	19.39	38.76
BertSumExtAbs	42.13	19.60	39.18

前3句

$R_1 R_2 \Rightarrow$ overlap
higher is better

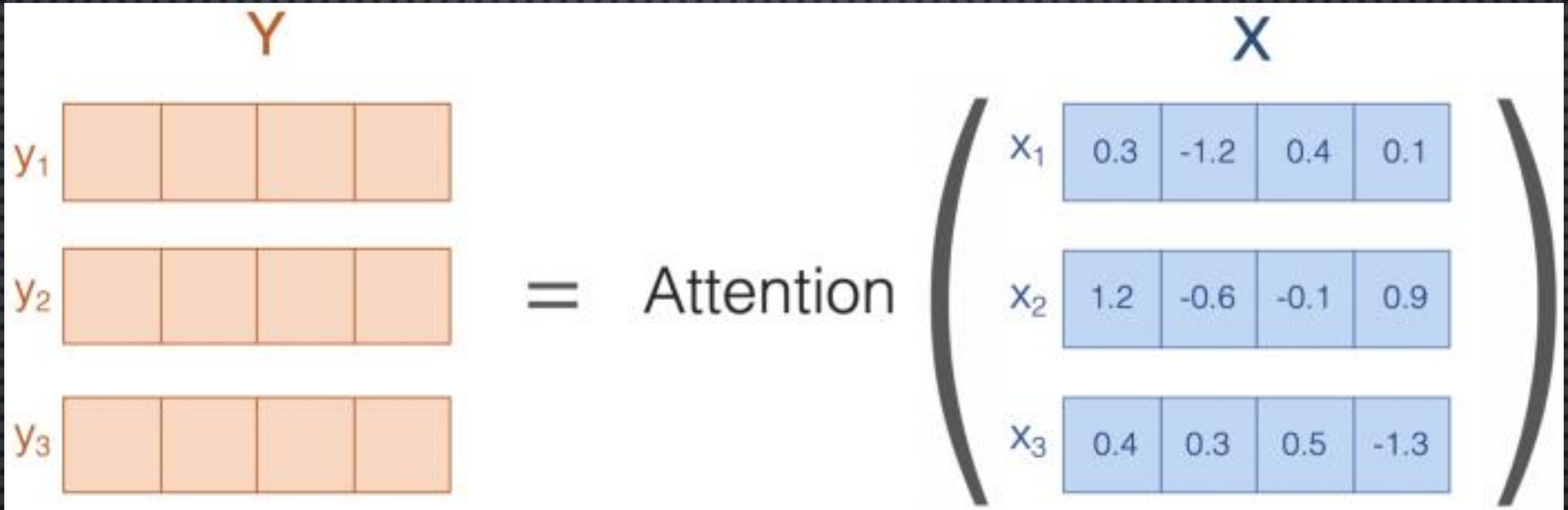
R_1: unigram
R_2: bigram

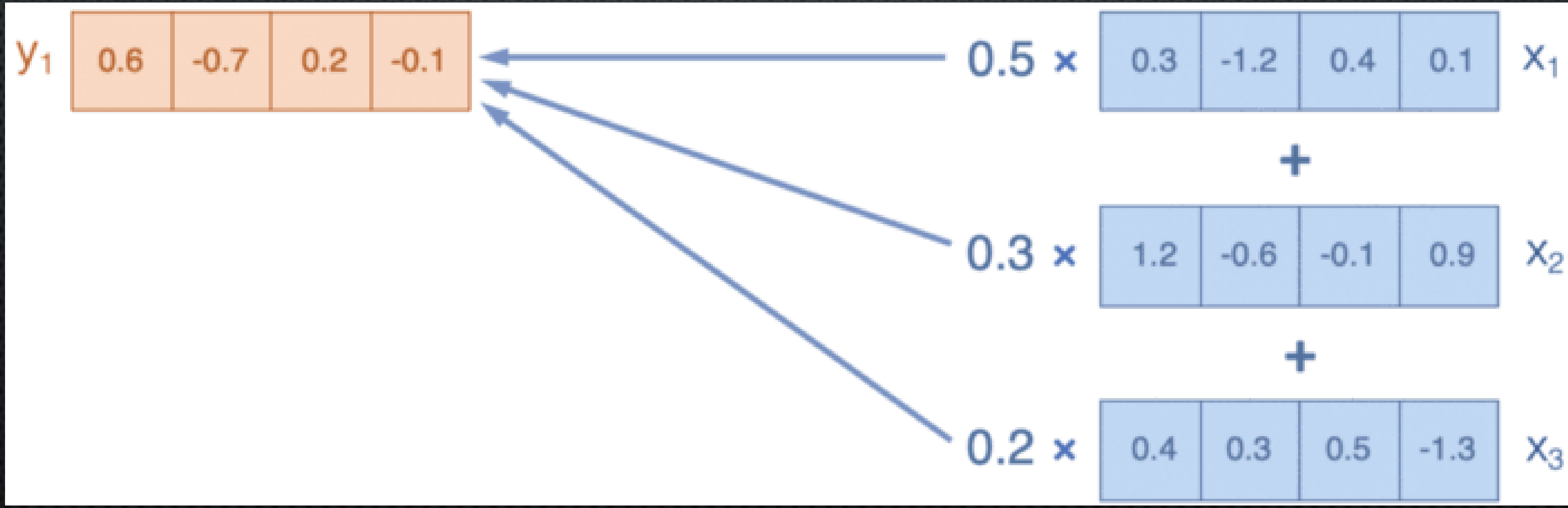
DEMO

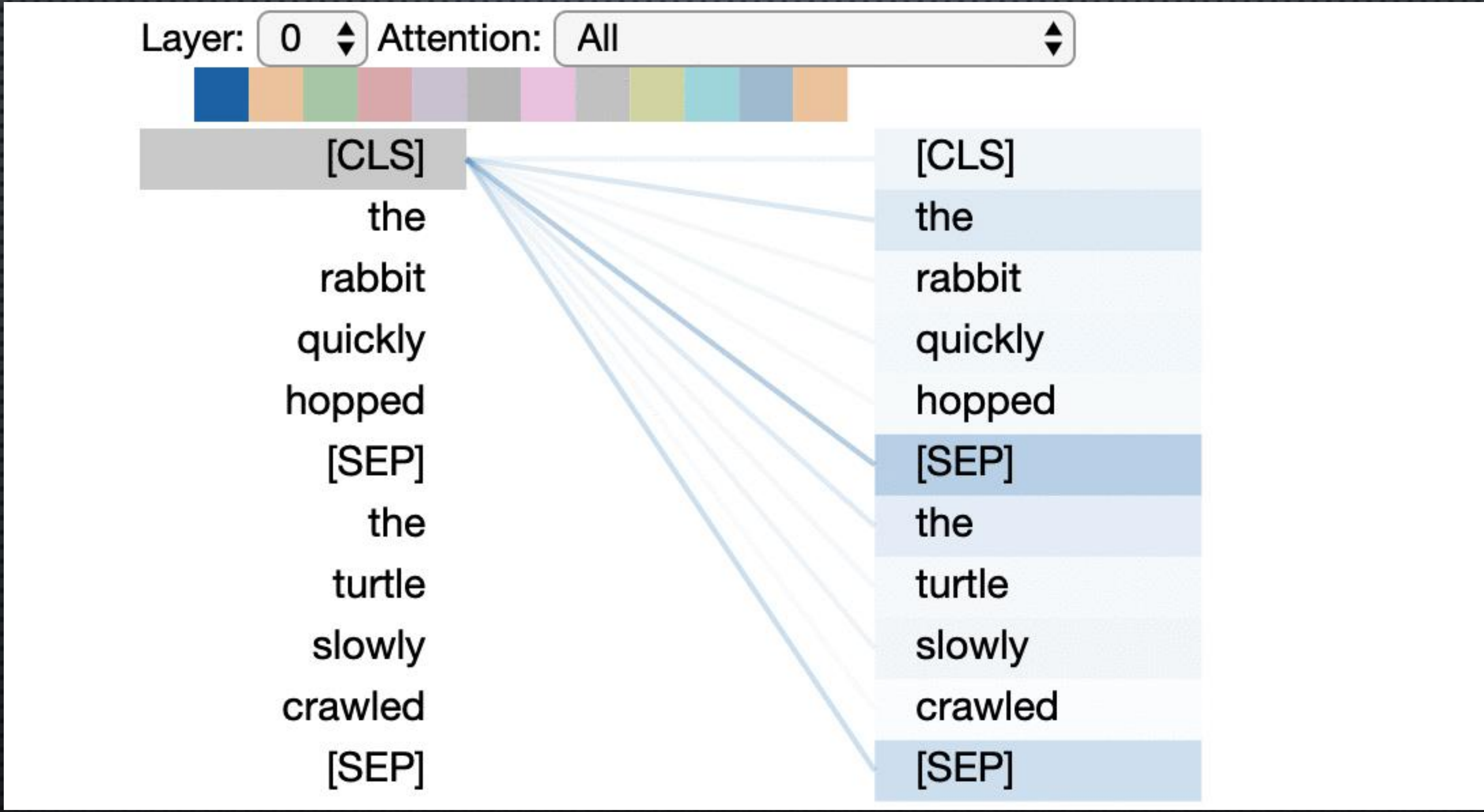
使用TRANSFORMER來判斷新聞摘要

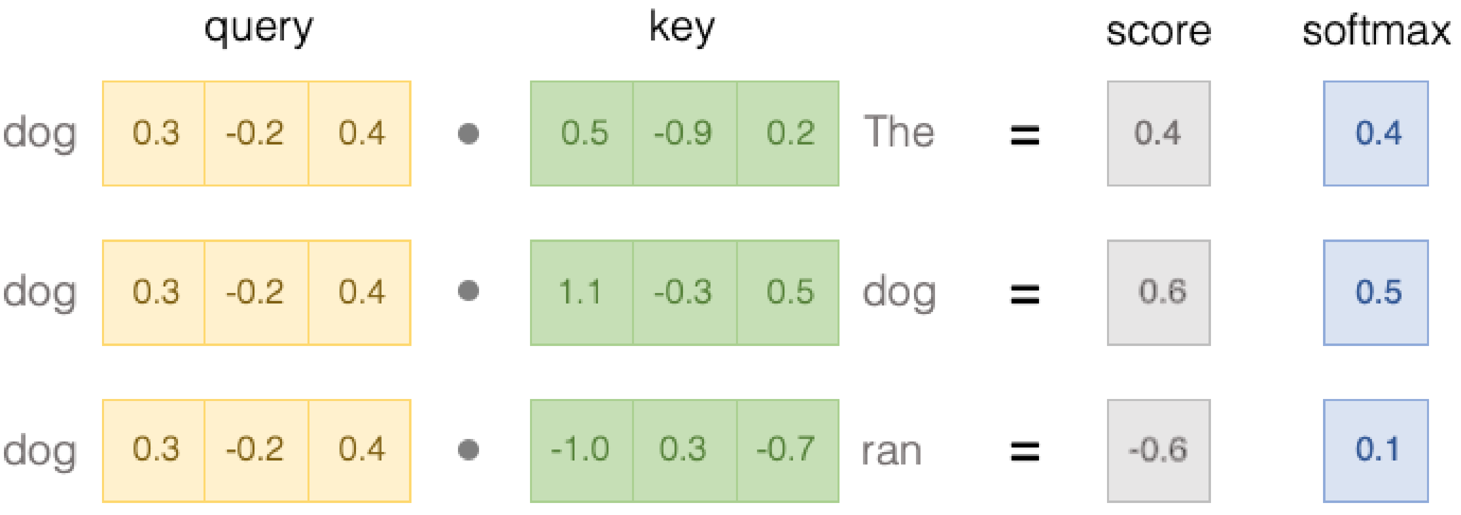
BERTVIZ

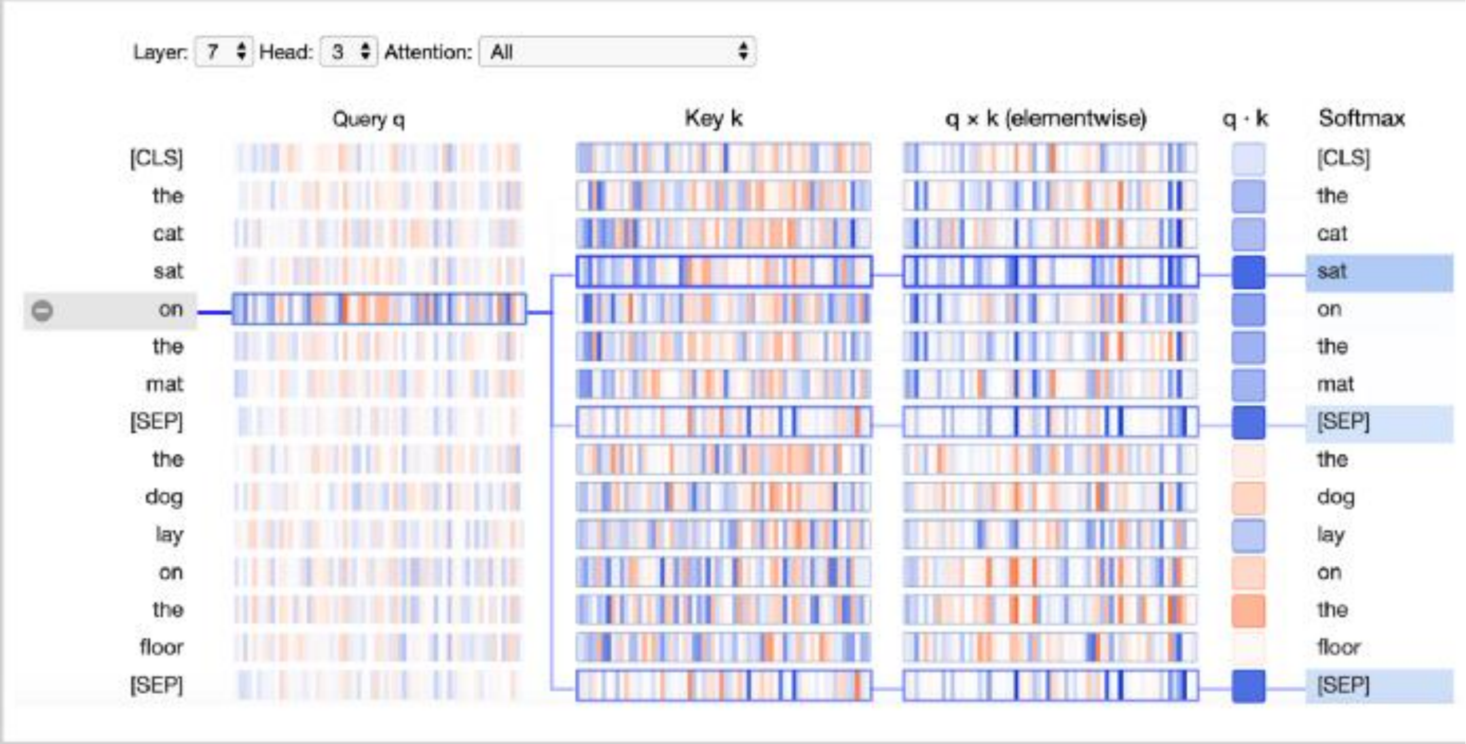
[JESSEVIG/BERTVIZ: BERTVIZ: VISUALIZE ATTENTION IN NLP MODELS (BERT, GPT2, BART, ETC.) (GITHUB.COM)]


Y
X
y1
y2
y3
= Attention
x1 0.3 -1.2 0.4 0.1
x2 1.2 -0.6 -0.1 0.9
x3 0.4 0.3 0.5 -1.3


y1 0.6 -0.7 0.2 -0.1
0.5 × 0.3 -1.2 0.4 0.1 x1
+
0.3 × 1.2 -0.6 -0.1 0.9 x2
+
0.2 × 0.4 0.3 0.5 -1.3 x3


Layer: 0
Attention: All
[CLS]
the
rabbit
quickly
hopped
[SEP]
the
turtle
slowly
crawled
[SEP]
[CLS]
the
rabbit
quickly
hopped
[SEP]
the
turtle
slowly
crawled
[SEP]





Query q: the query vector q encodes the word on the left that is paying attention, i.e. the one that is "querying" the other words. In the example above, the query vector for "on" (the selected word) is highlighted.

Key k: the key vector k encodes the word on the right to which attention is being paid. The key vector and the query vector together determine a compatibility score between the two words.

q×k (elementwise): the elementwise product between the query vector of the selected word and each of the key vectors. This is a precursor to the dot product (the sum of the elementwise product) and is included for visualization purposes because it shows how individual elements in the query and key vectors contribute to the dot product.

q·k: the scaled dot product (see above) of the selected query vector and each of the key vectors. This is the unnormalized attention score.

Softmax: the softmax of the scaled dot product. This normalizes the attention scores to be positive and sum to one.

DEMO

使用BERTVIZ視覺化ATTENTION