CS772: Deep Learning for Natural Language Processing (DL-NLP)

Self Attention, Vaswani et al 2018

Pushpak Bhattacharyya

Computer Science and Engineering Department

IIT Bombay

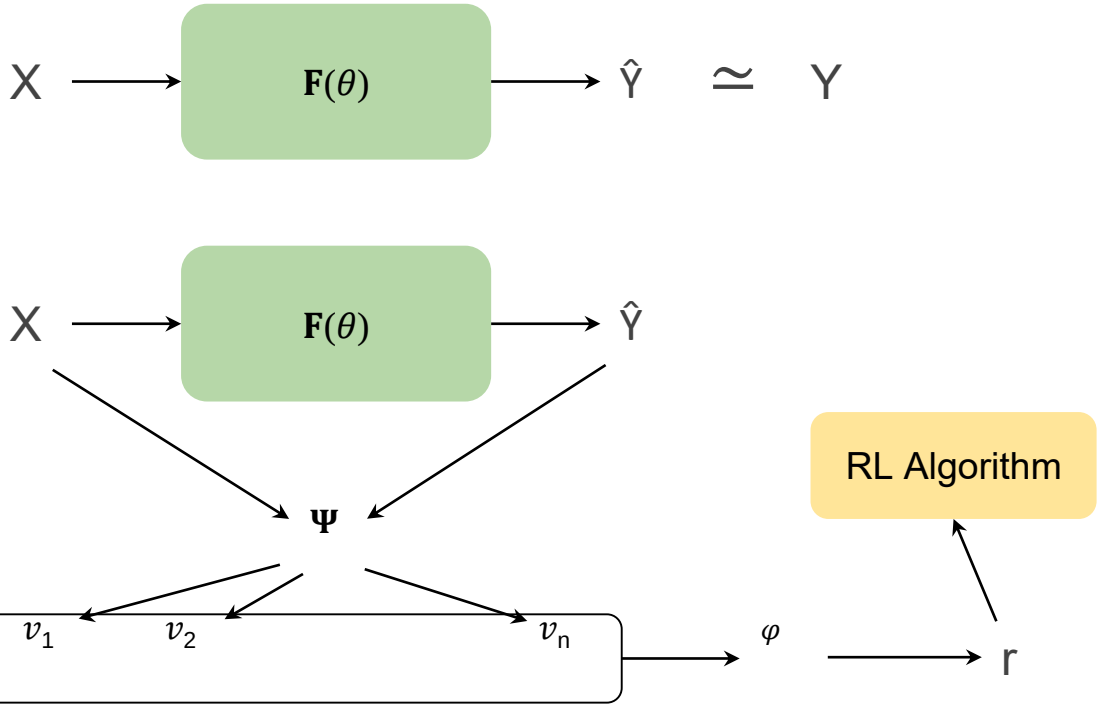
Week 13 of 1apr24

(Thursday lecture was on elastic LLMs by Dr. Prateek Jain, Google Research)

1-slide recap

$$\tilde{e} = \underset{e \in e^*}{\operatorname{argmax}}\, p(e|f) = \underset{e \in e^*}{\operatorname{argmax}}\, p(f|e)p(e)$$

SMT



Domain specific RLHF

Self attention

Self attention in real life… (1/2)

- Coreference Resolution
- **Sentence-1 (S_1):** *The$_1$ cat$_2$ could$_3$ not$_4$ climb$_5$ the$_6$ wall$_7$ because$_8$ it$_9$ was$_{10}$ too$_{11}$ steep$_{12}$ and$_{13}$ smooth$_{14}$.$_{15}$*
- **Sentence-2 (S_2):** *The$_1$ cat$_2$ could$_3$ not$_4$ climb$_5$ the$_6$ wall$_7$ because$_8$ it$_9$ was$_{10}$ too$_{11}$ weak$_{12}$ and$_{13}$ wounded$_{14}$.$_{15}$*
  - **S_1: Coref(9)=7**
  - **S_2: Coref(9)=2**

Self attention in real life… (2/2)

- Semantic Role Labelling (SRL)

- **Sentence-3 (S_3):** I_1 *promised*$_2$ *him*$_3$ *to*$_4$ *give*$_5$ a_6 *party*$_7$.$_8$
- **Sentence-4 (S_4):** I_1 *forced*$_2$ *him*$_3$ *to*$_3$ *give*$_4$ a_5 *party*$_7$.$_8$
  - **S_3: agent(5)=1**
  - **S_4: agent(5)=3**

Probing through translation (1/2)

- I promised him to give a party
- I forced him to give a party
- The cat could not climb the wall because it was too steep
- The cat could not climb the wall because it was too weak

- मैंने उससे पार्टी देने का वादा किया
- मैंने उस पर पार्टी देने के लिए दबाव डाला
- बिल्ली दीवार पर नहीं चढ़ सकी क्योंकि वह बहुत खड़ी थी
- बिल्ली दीवार पर नहीं चढ़ सकी क्योंकि वह बहुत कमज़ोर थी

Google Translate

Probing through translation (2/2)

- The child could not climb the wall because it was too steep
- The child could not climb the wall because it was too small
- The child could not climb the wall because it was too weak

- बच्चा दीवार पर नहीं चढ़ सका क्योंकि वह बहुत खड़ी थी
- बच्चा दीवार पर नहीं चढ़ सका क्योंकि वह बहुत छोटी थी
- बच्चा दीवार पर नहीं चढ़ सका क्योंकि वह बहुत कमजोर थी

Google Translate

IIT Translator: Probing through translation (1/2)

- I promised him to give a party
- I forced him to give a party
- The cat could not climb the wall because it was too steep
- The cat could not climb the wall because it was too weak

- मैंने उनसे पार्टी देने का वादा किया था।
- मैंने उसे एक पार्टी देने के लिए मजबूर किया
- बिल्ली दीवार पर चढ़ नहीं सकती थी क्योंकि यह बहुत खड़ी थी।
- बिल्ली दीवार पर चढ़ नहीं सकती थी क्योंकि वह बहुत कमजोर थी

IIT Translator: Probing through translation (2/2)

- The child could not climb the wall because it was too steep
- The child could not climb the wall because it was too small
- The child could not climb the wall because it was too weak

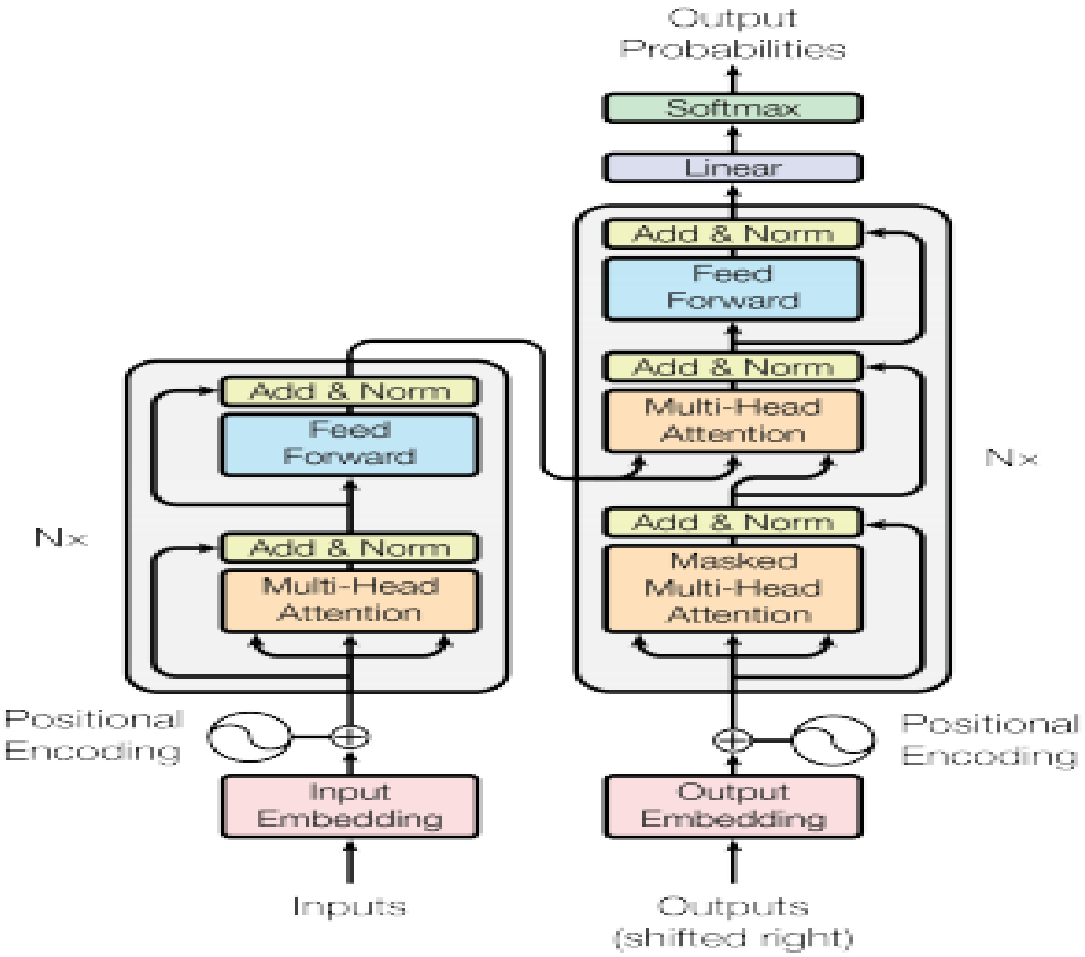
- बच्चा दीवार पर चढ़ नहीं सका क्योंकि यह बहुत खड़ी थी।
- बच्चा दीवार पर चढ़ नहीं सका क्योंकि वह बहुत छोटा था
- बच्चा दीवार पर नहीं चढ़ सका क्योंकि वह बहुत कमजोर था

Digression: Linguistic Probe

- How do we know ‘cat’ is a noun?
- ‘cat’ can replace ‘dog’- which is a NOUN (known from another source)- in identical syntactic environment
- *I saw a dog* ←→ *I saw a cat*

Vaswani et al 2018

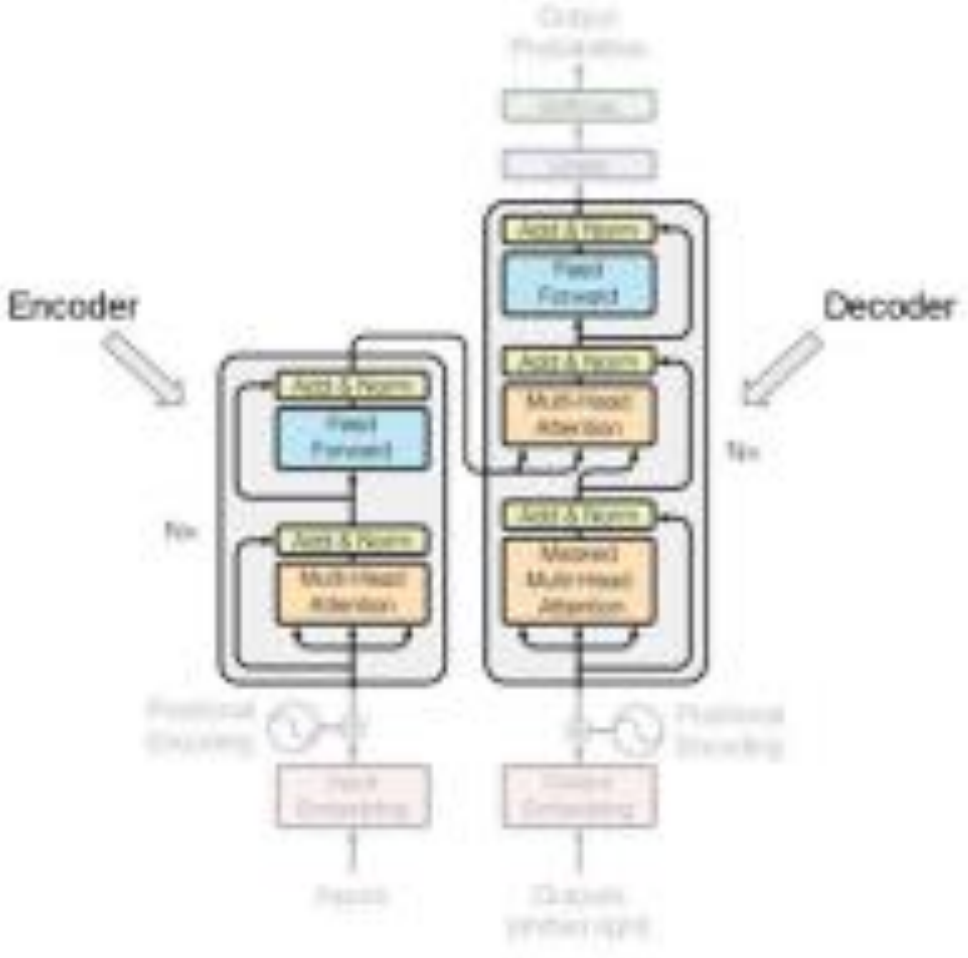
A classic diagram and a classic paper



Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." NeurIPS (2017).

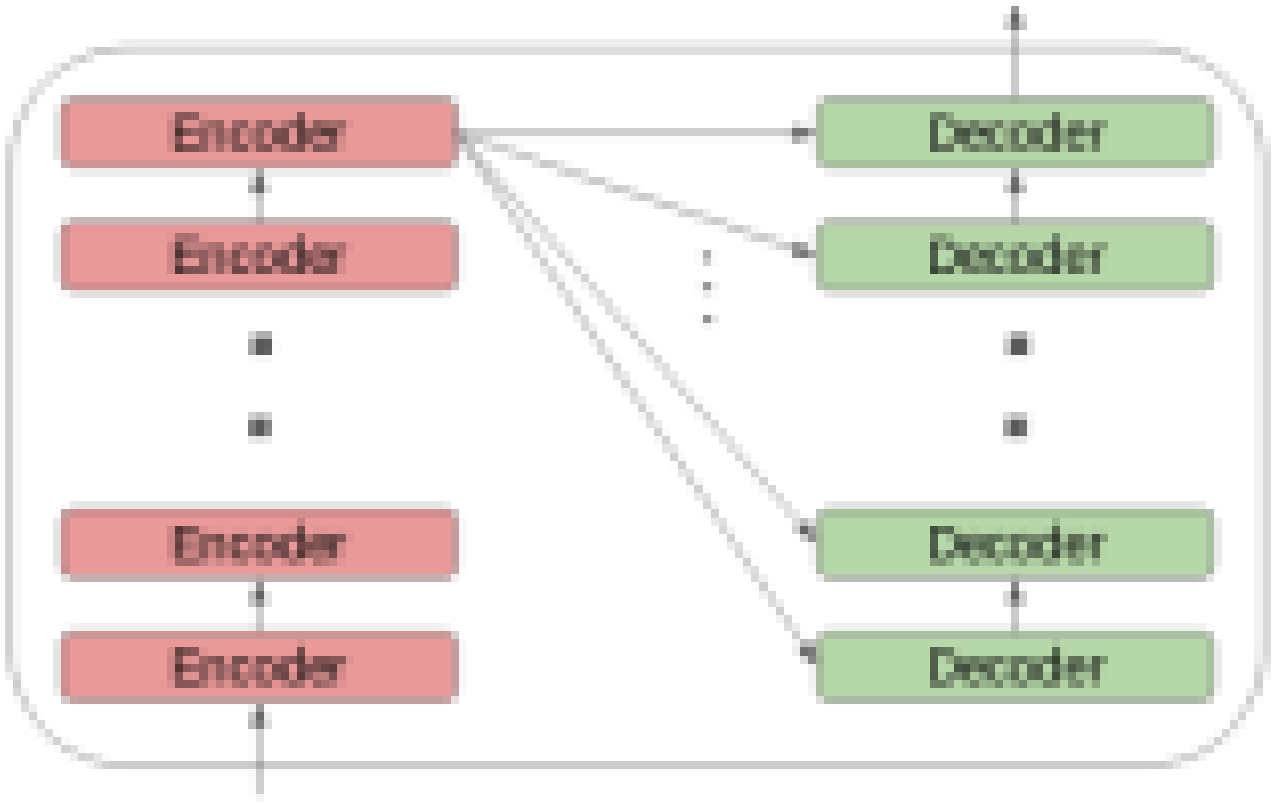
http://nlp.seas.harvard.edu/2018/04/03/attention.html
http://jalammar.github.io/illustrated-transformer/

The transformer

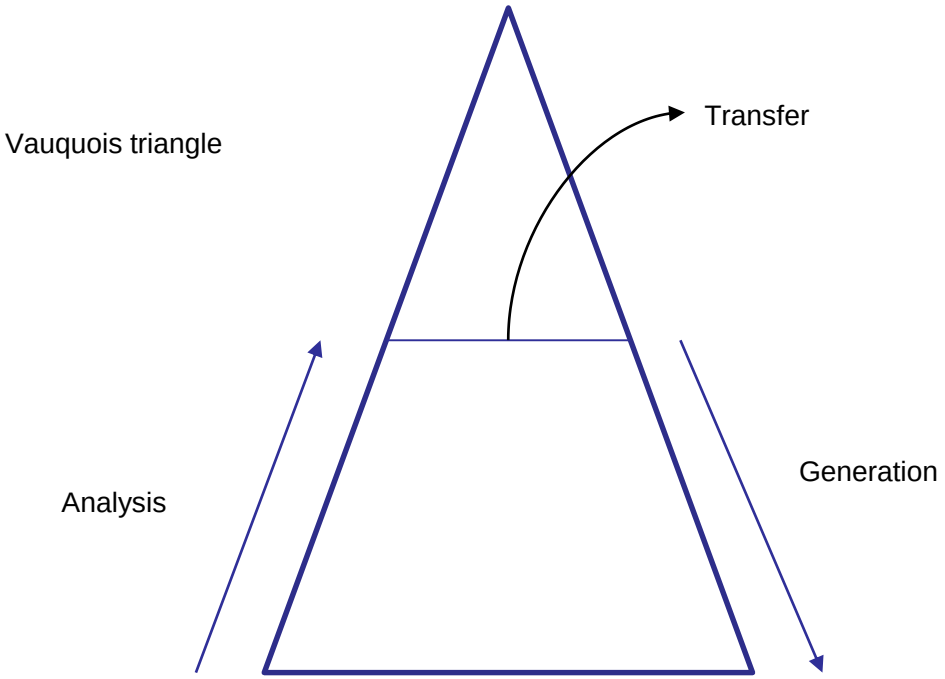


Nx means N times; N=6 conventionally

Encoder decoder interaction



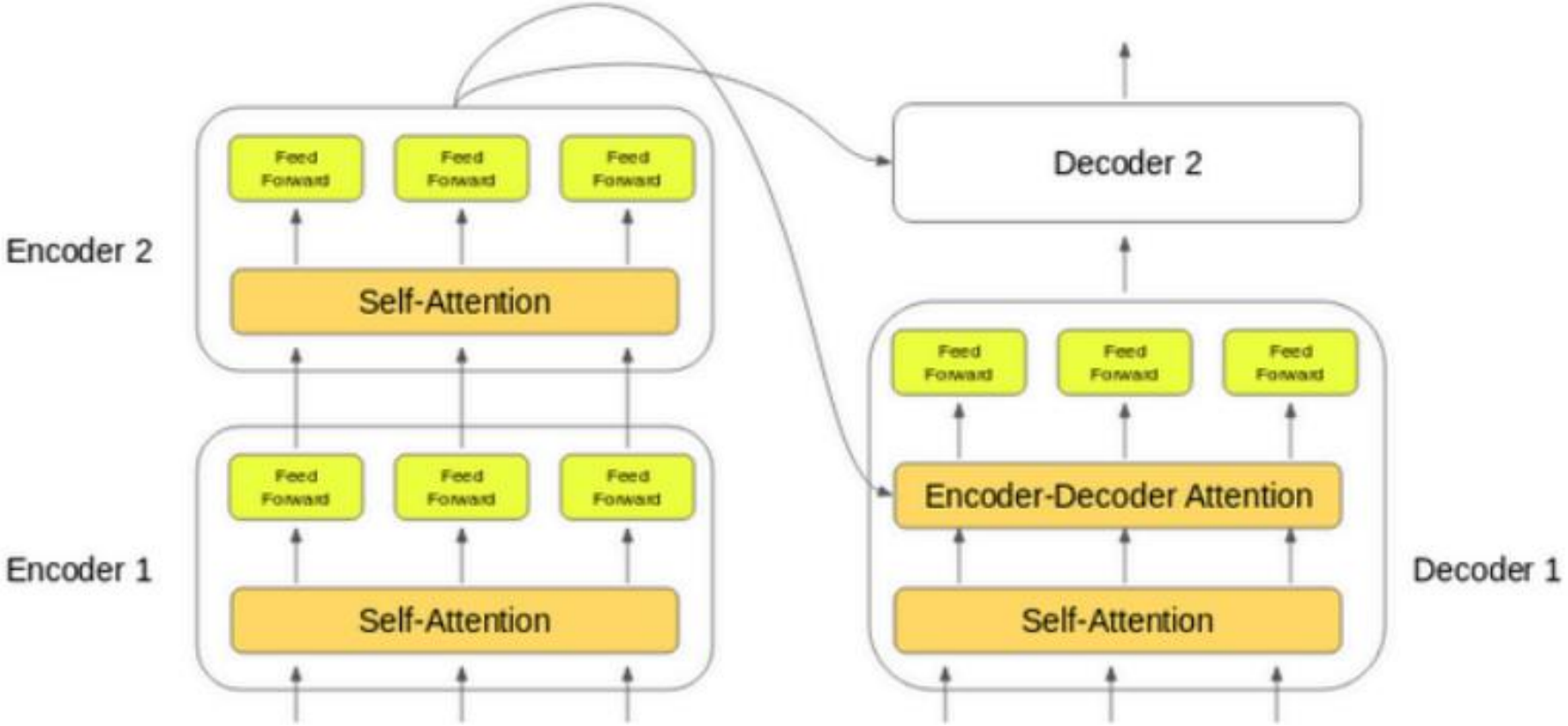
Vauquois triangle



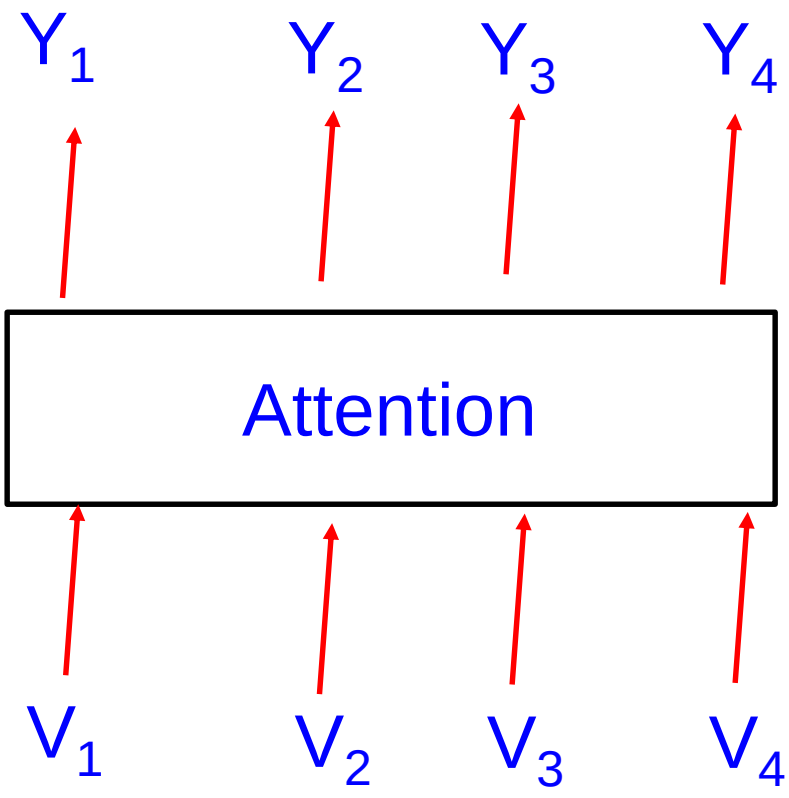
Stages of NL Generations (NLG)

- Vocab generation
- Morph and function word generation
- Syntax planning
- Example
  - Input: Peter slept early
  - Vocab: पीटर सो जल्दी (Peter so jaldii)
  - Morph and function words: पीटर सोया जल्दी (Peter soya jaldii)
  - Syntax planning: पीटर जल्दी सोया (Peter jaldii soya)

Self Attention as part of the architecture



Self Attention Block



Bank of the river

Word Embedding and Contextual Word Embedding

- Consider the phrase "*bank of the river*"
- Word embeddings of '*bank*', '*of*', '*the*', '*river*': V_1, V_2, V_3, V_4
- Now create a 'score' vector S_i for each word vector
- S_1: ($V_1.V_1$, $V_1.V_2$, $V_1.V_3$, $V_1.V_4$)
- Similarly, S_2, S_3, S_4

S-matrix

$$S = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

S-scaled matrix

$$S - scaled = \frac{1}{\sqrt{d_k}} \times \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix}$$

W-matrix

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix}$$

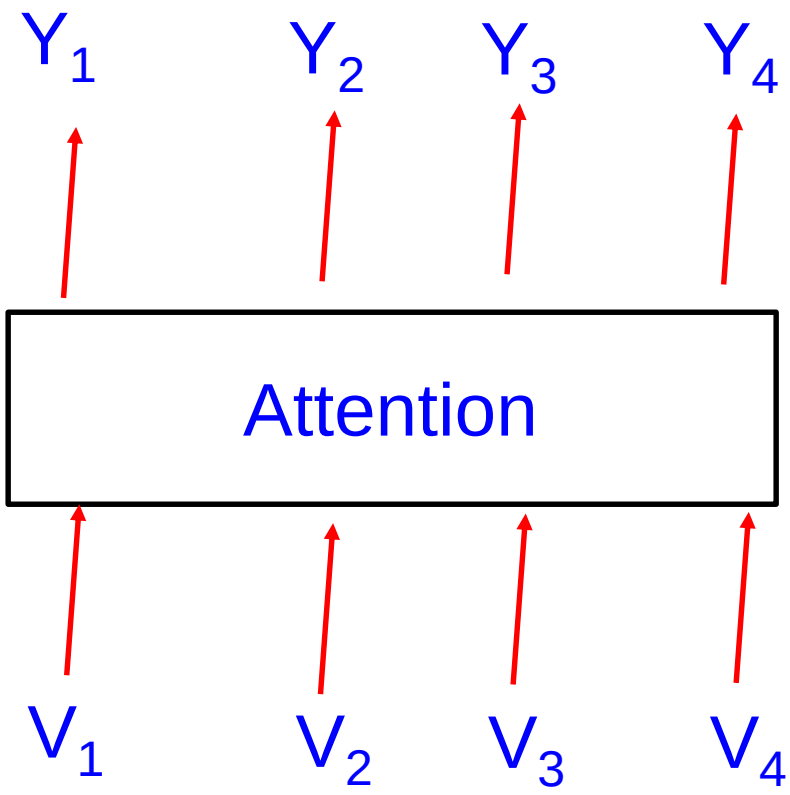
$$W_i - vector = soft\max\left(\frac{S_i - vector}{\sqrt{d_k}}\right)$$

Y-matrix

$$Y = \begin{bmatrix} y_{11}\, y_{12}\, y_{13}\, y_{14} \\ y_{21}\, y_{22}\, y_{23}\, y_{24} \\ y_{31}\, y_{32}\, y_{33}\, y_{34} \\ y_{41}\, y_{42}\, y_{43}\, y_{44} \end{bmatrix}$$

$$Y_i - vector = w_{11}.V_1 + w_{12}.V_2 + w_{13}.V_3 + w_{14}.V_4$$

Attention Block



Bank of the river

Deeper dive into attention

Self Attention (1/3)

- Create 3 vectors using the input embeddings (x).
  - Query (q)
  - Key (k)
  - Value (v)
- Obtain these vectors by matrix multiplication with the weight matrix W^Q, W^K, W^V which are the parameters of the self attention module
- These matrices are learnable

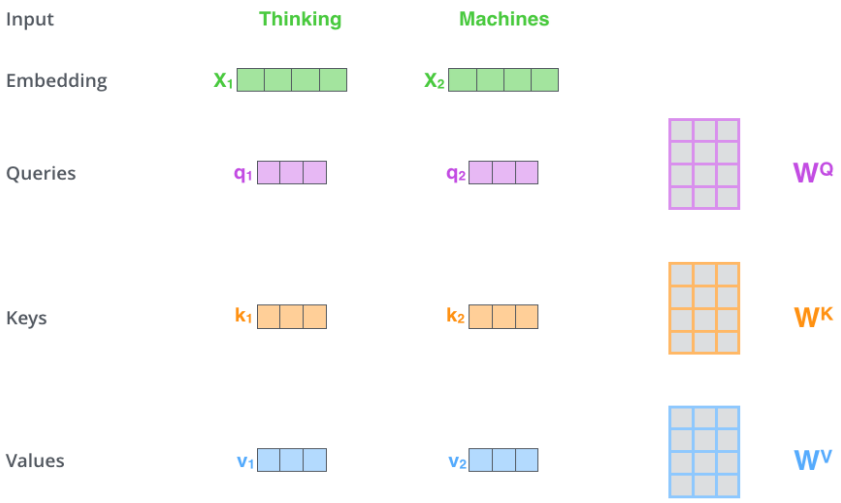


Image Source: The Illustrated Transformer, https://jalammar.github.io/illustrated-transformer/

Self Attention (2/3)

Take the dot product of the query(q) vector of current word with the key(k) vector of each input word.

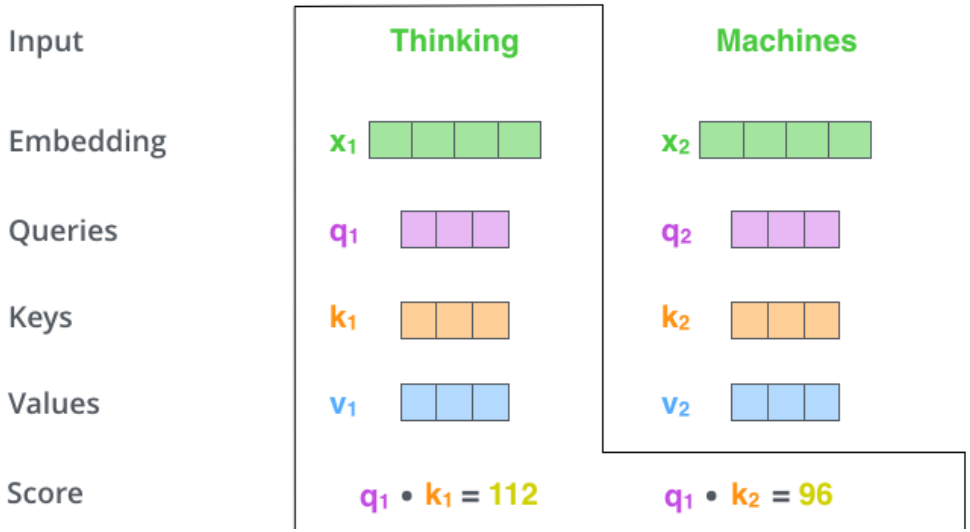


Image Source: The Illustrated Transformer, https://jalammar.github.io/illustrated-transformer/

Self Attention (3/3)

- Scale the scores by dividing the scores by d_k and then we perform the softmax operation on the scores.
- Weight the value (v) vectors by multiplying the vectors with the corresponding scores of that position.
- Compute the weighted sum of the value (v) vectors which forms the output of self attention layer.

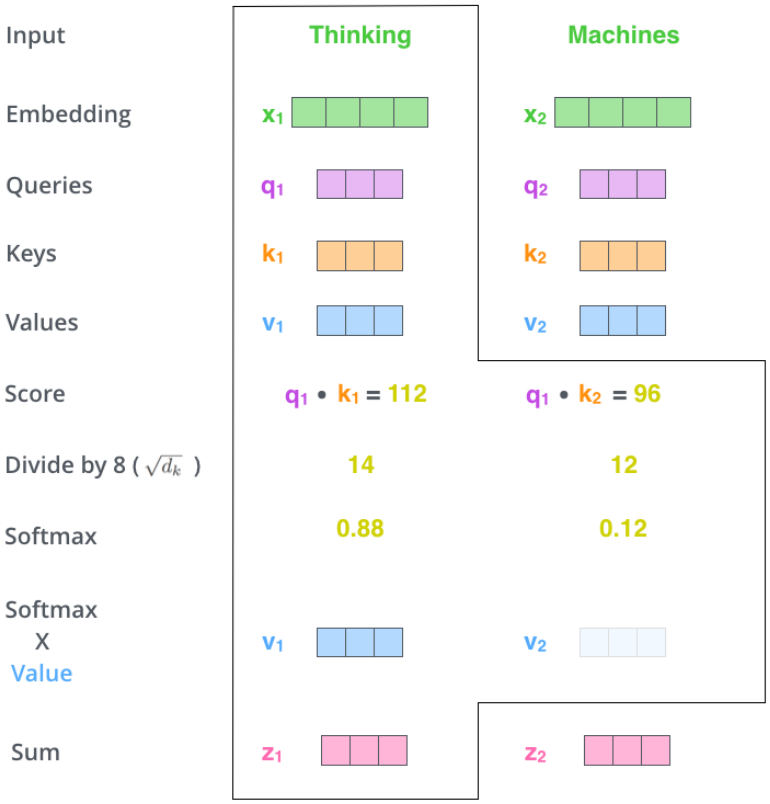


Image Source: The Illustrated Transformer, https://jalammar.github.io/illustrated-transformer/