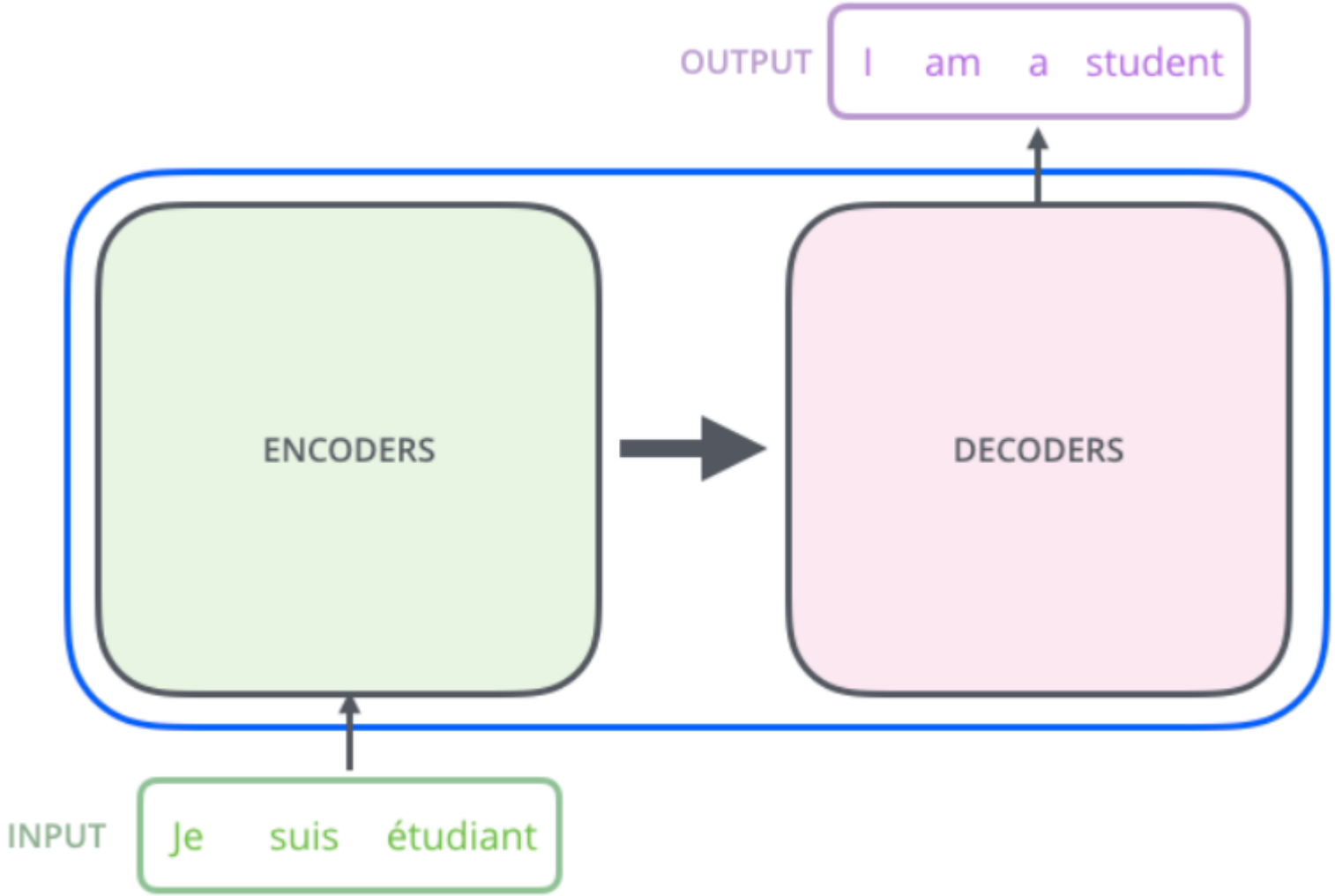
Group Meeting

刘广熠

Contents

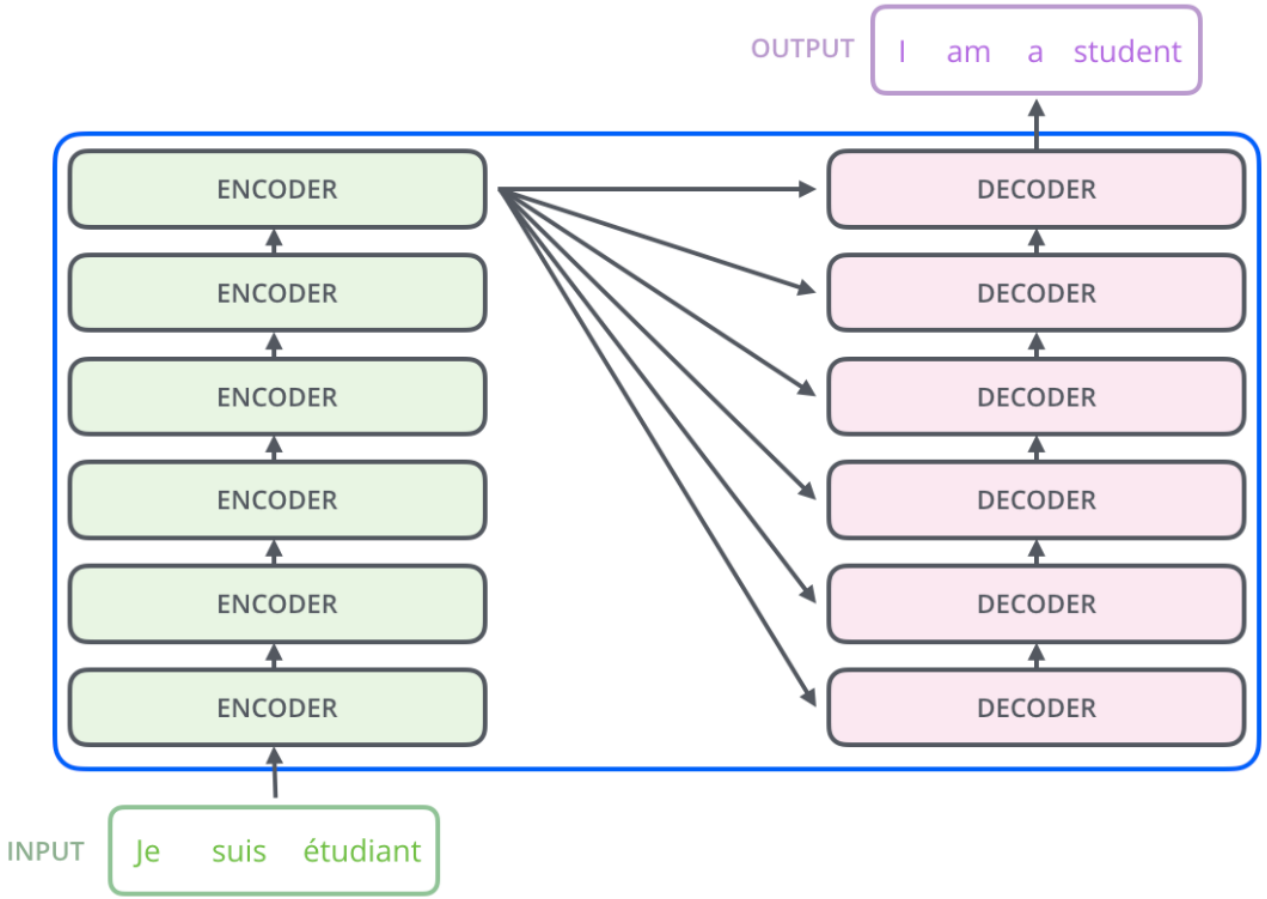
- Review of Transformer
- Review of BERT
- VLBERT @ ICLR 2020

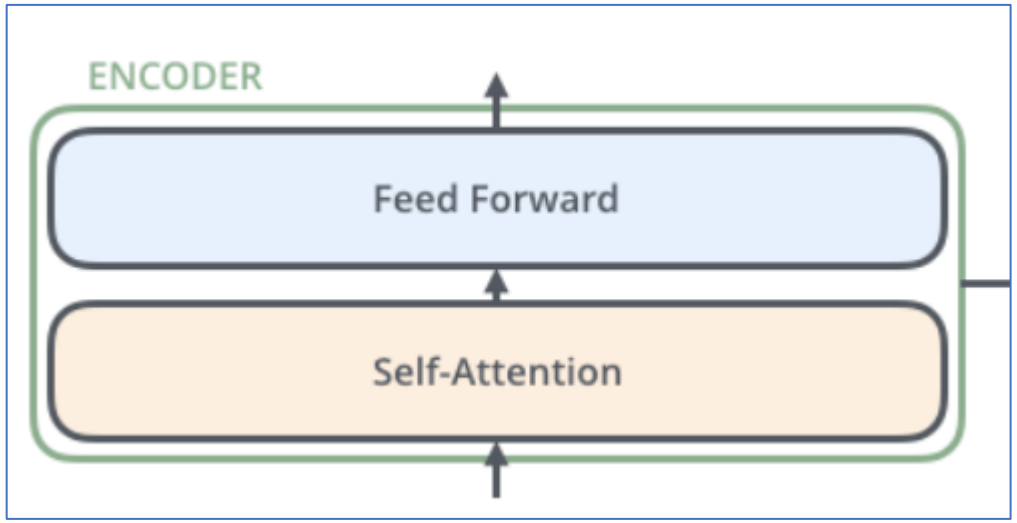
Review of Transformer

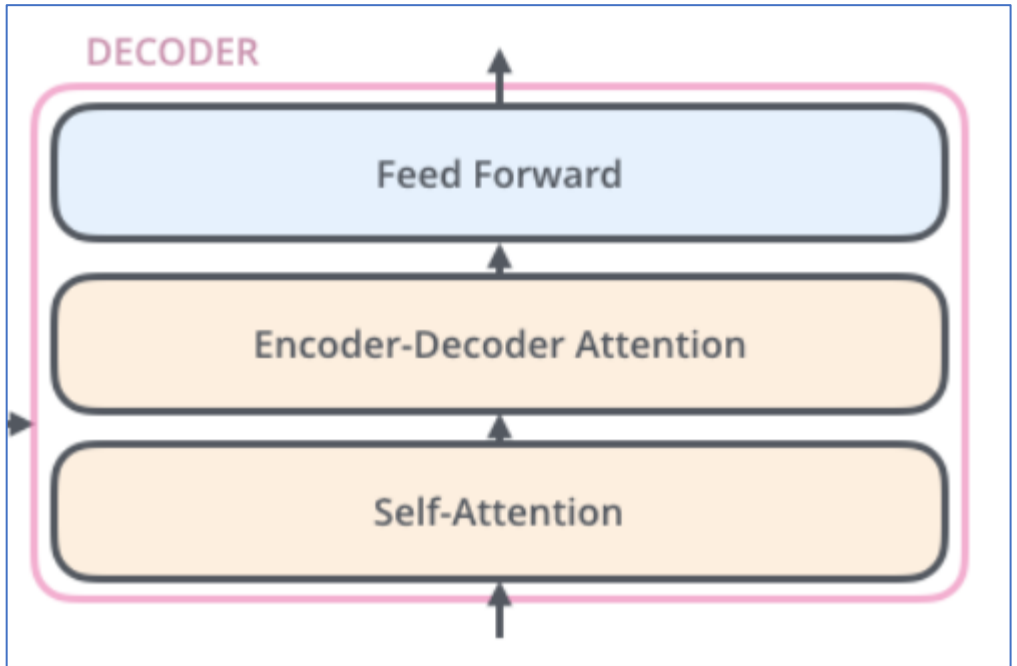


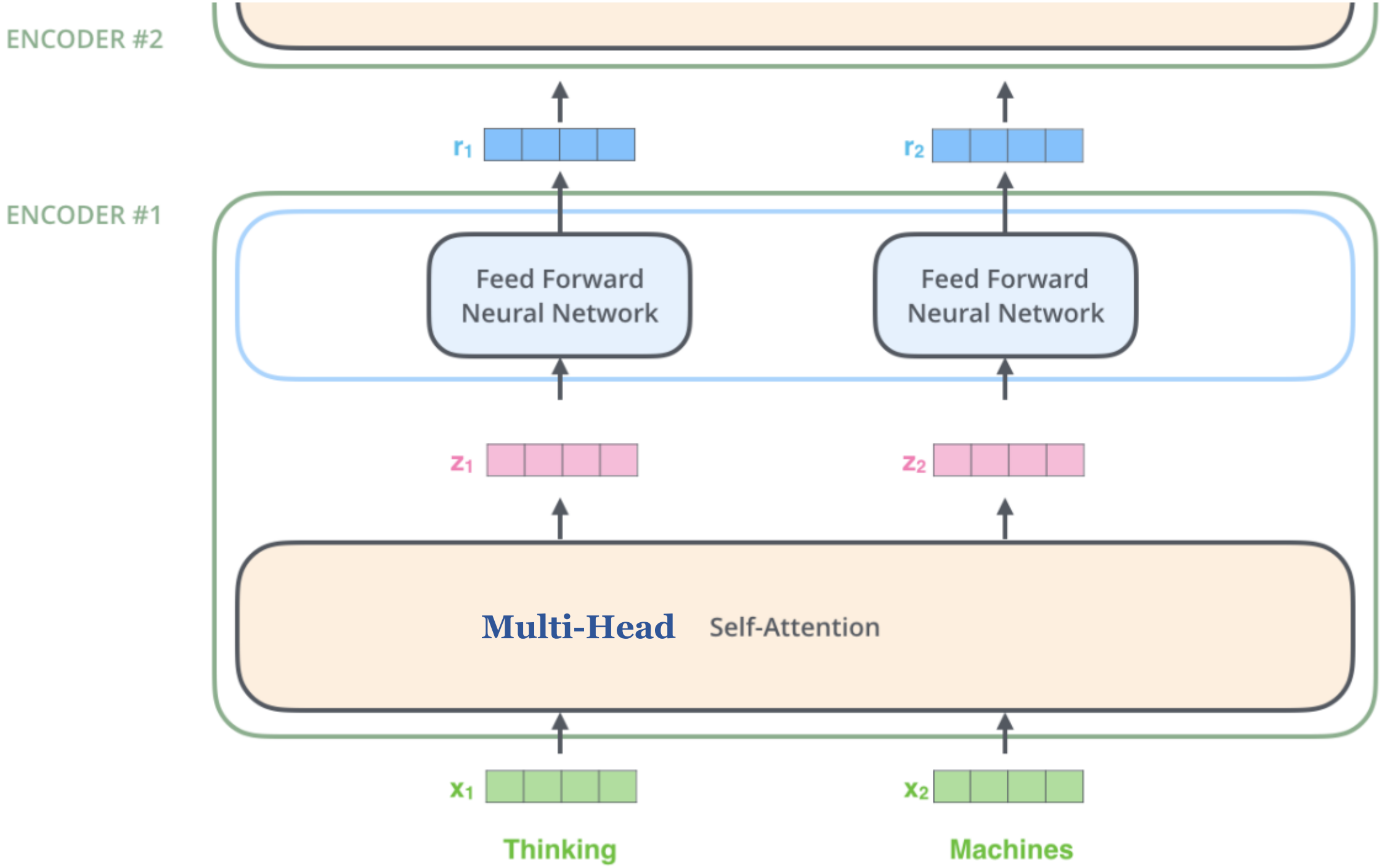
Reference: [The Illustrated Transformer]

Transformer

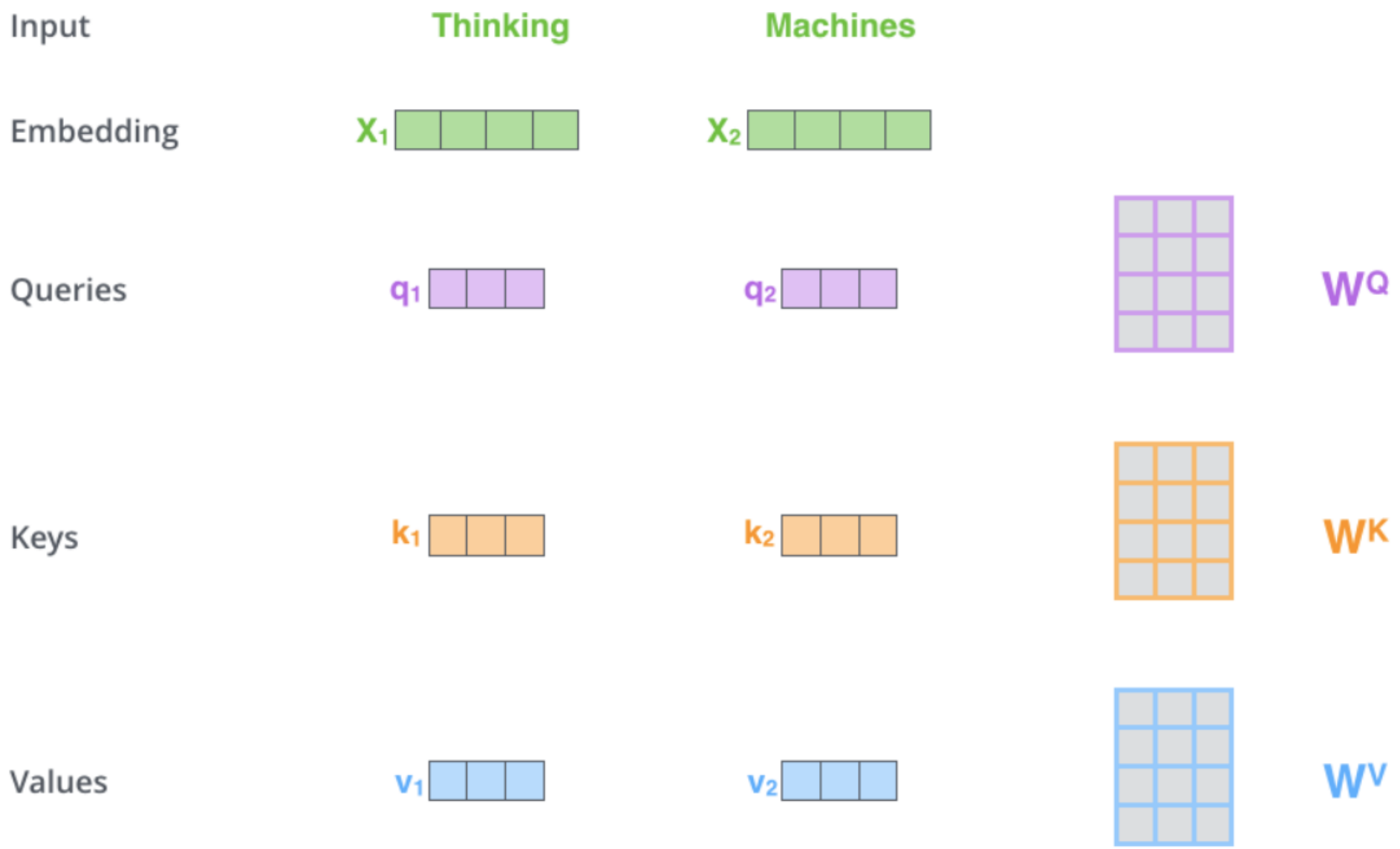


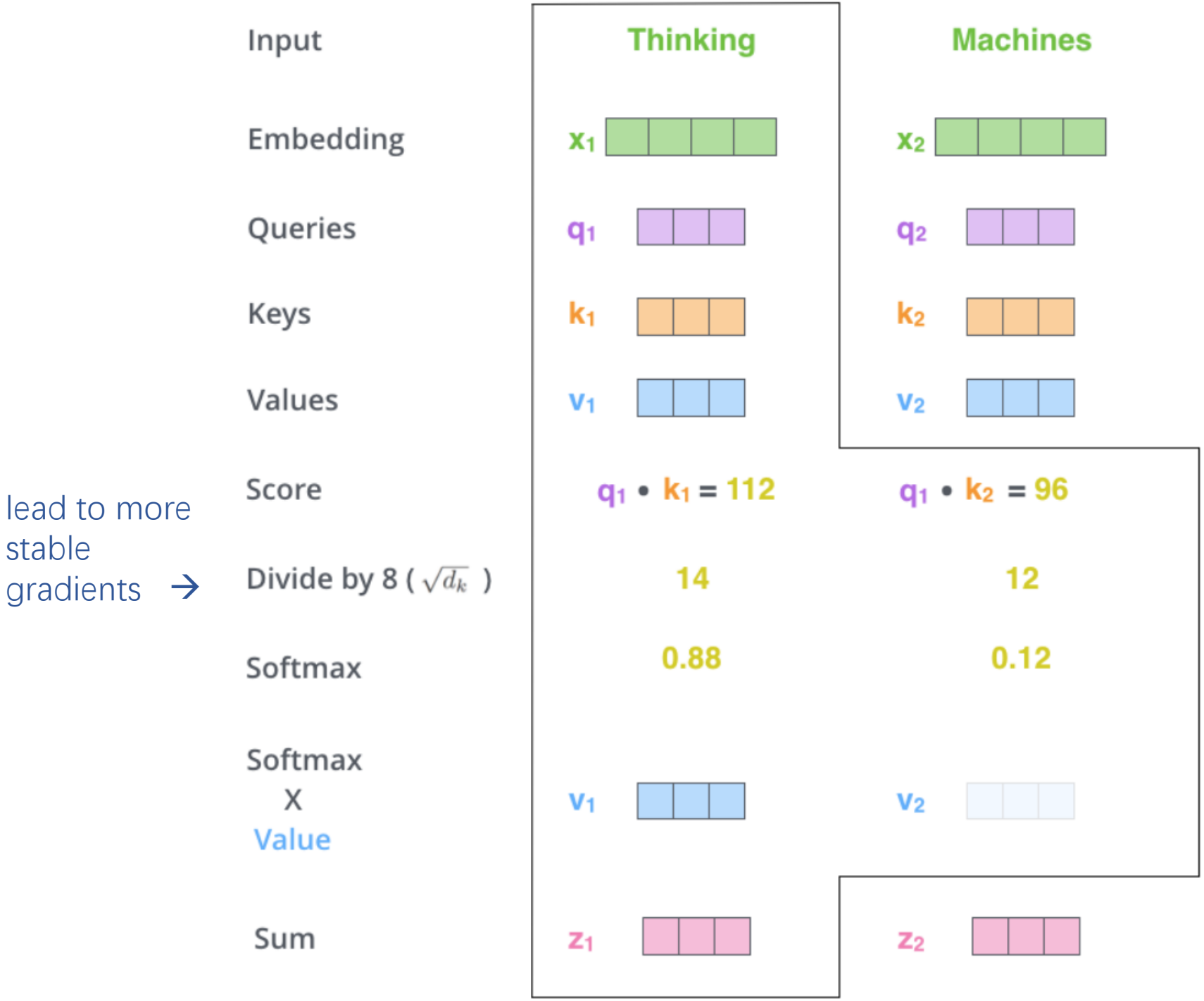


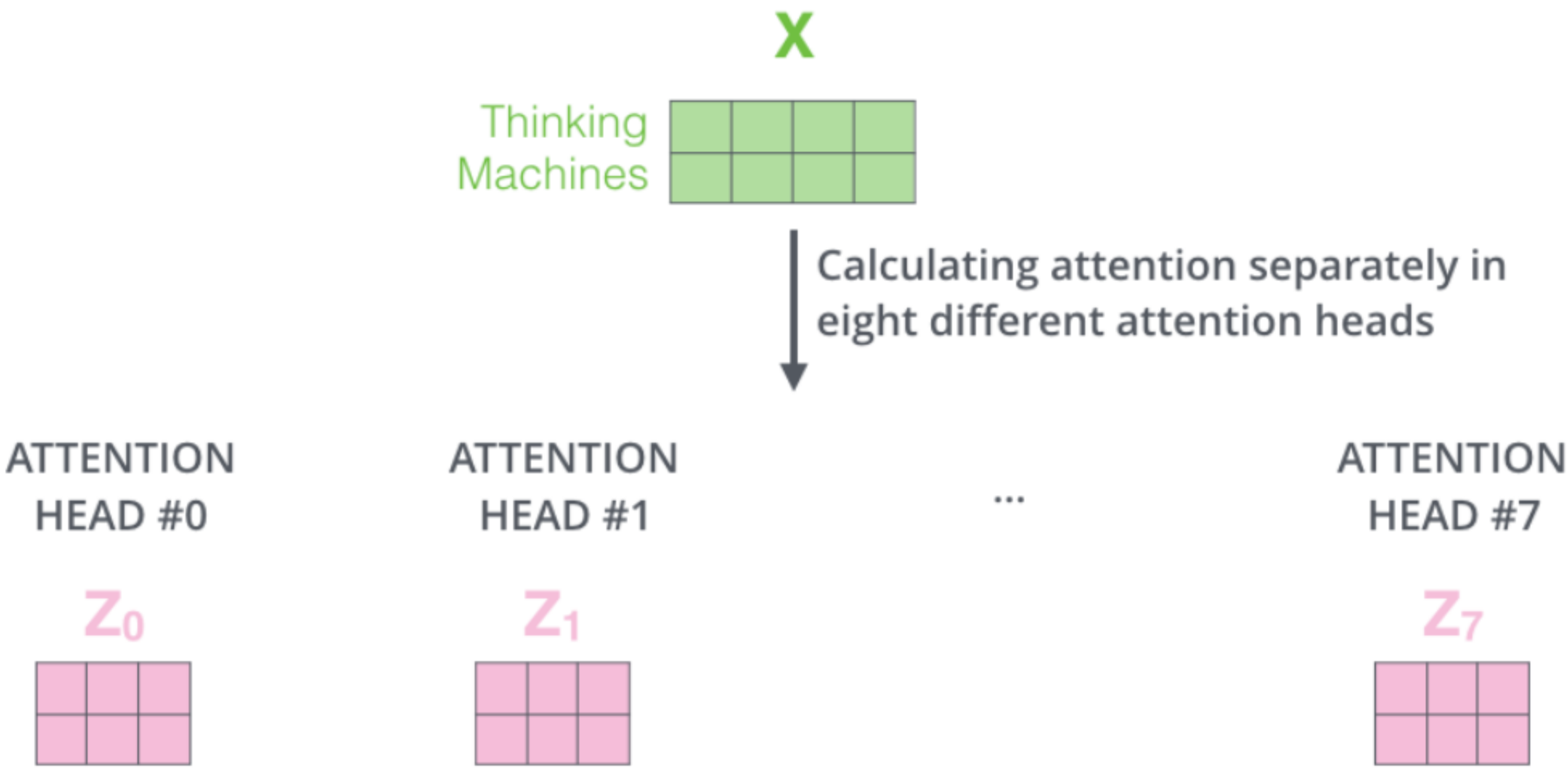



ENCODER #2
ENCODER #1
r1
r2
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Multi-Head Self-Attention
x1
x2
Thinking
Machines

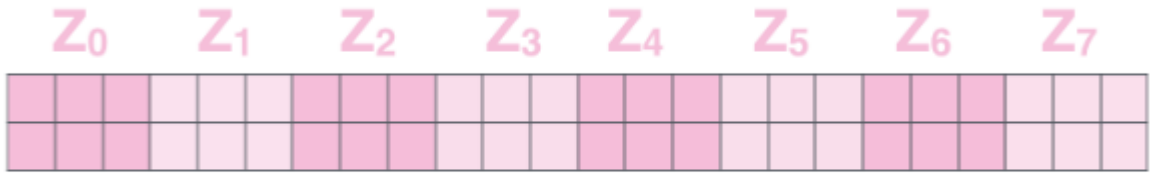
Self-Attention




Input
Thinking
Machines
Embedding
x_1
x_2
Queries
q_1
q_2
Keys
k_1
k_2
Values
v_1
v_2
Score
$q_1 \bullet k_1 = 112$
$q_1 \bullet k_2 = 96$
lead to more stable gradients →
Divide by 8 ($\sqrt{d_k}$)
14
12
Softmax
0.88
0.12
Softmax X Value
v_1
v_2
Sum
z_1
z_2


X
Thinking
Machines
Calculating attention separately in eight different attention heads
ATTENTION HEAD #0
ATTENTION HEAD #1
...
ATTENTION HEAD #7
Z0
Z1
Z7

1) Concatenate all the attention heads

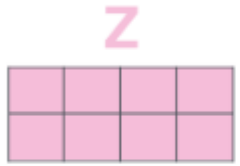


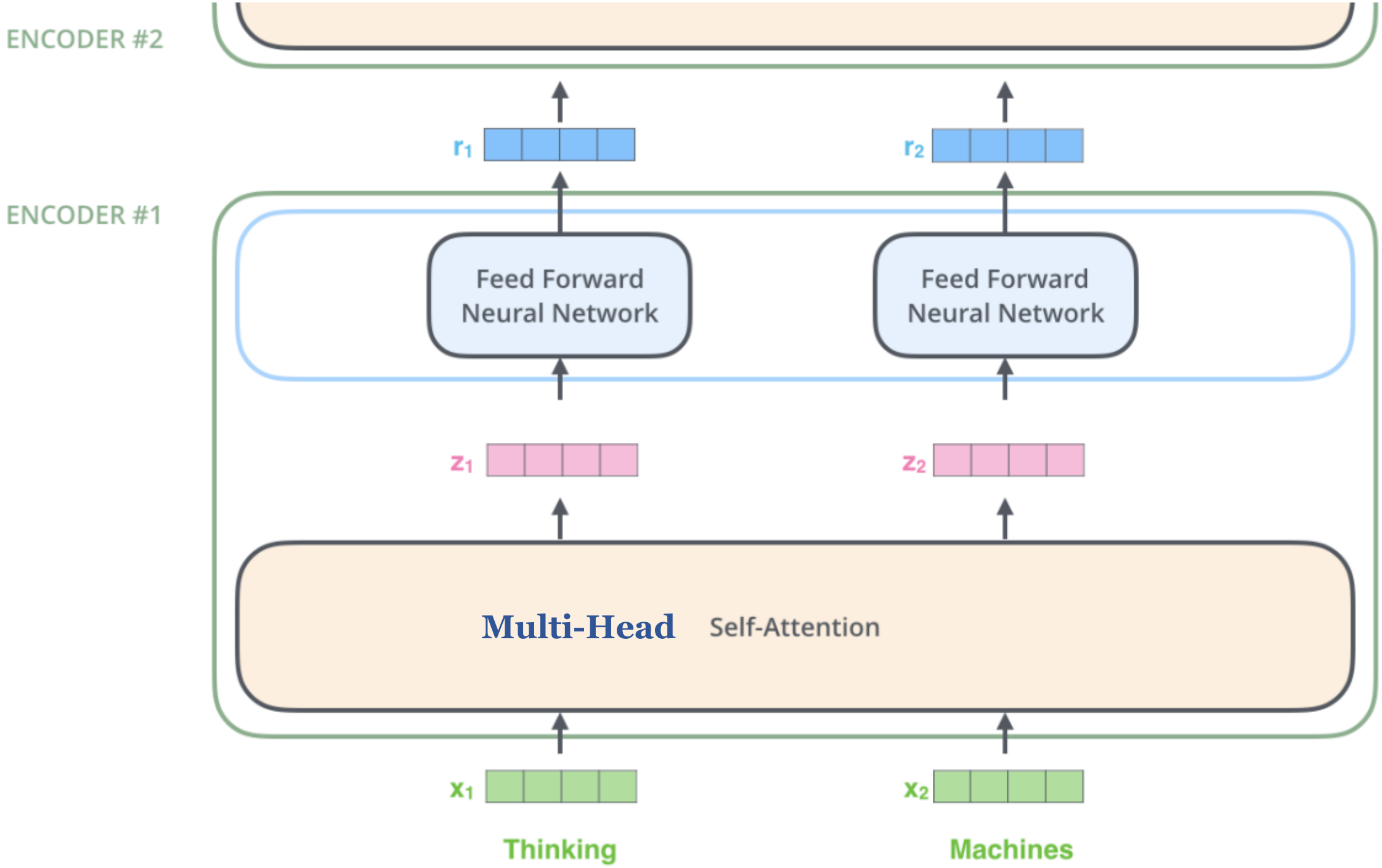
2) Multiply with a weight matrix W^O that was trained jointly with the model

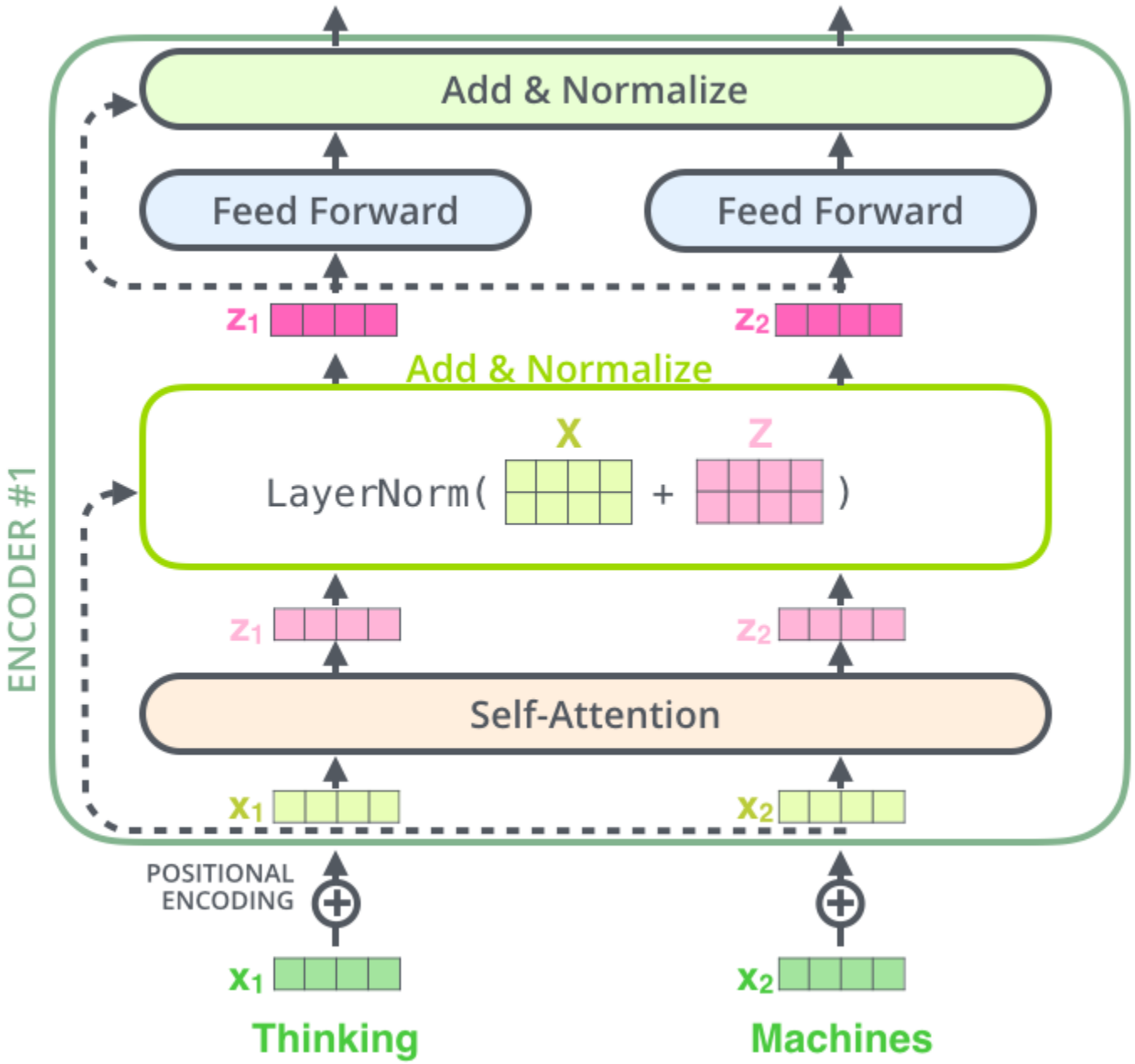
X

3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

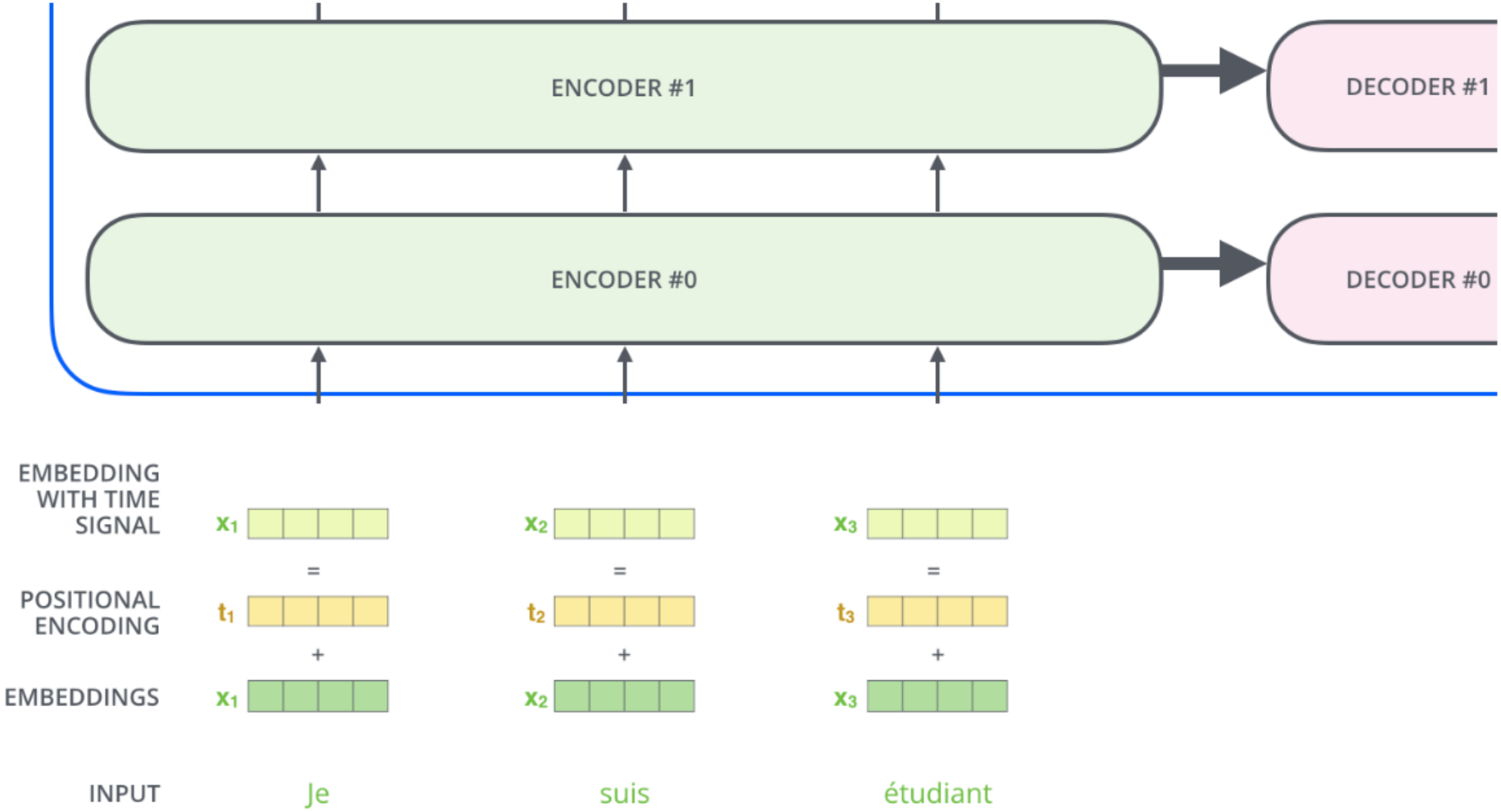
=




ENCODER #2
ENCODER #1
r1
r2
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Multi-Head Self-Attention
x1
x2
Thinking
Machines


Add & Normalize
Feed Forward
Feed Forward
z1
z2
Add & Normalize
X
Z
LayerNorm(+)
z1
z2
Self-Attention
x1
x2
ENCODER #1
POSITIONAL ENCODING
x1
x2
Thinking
Machines

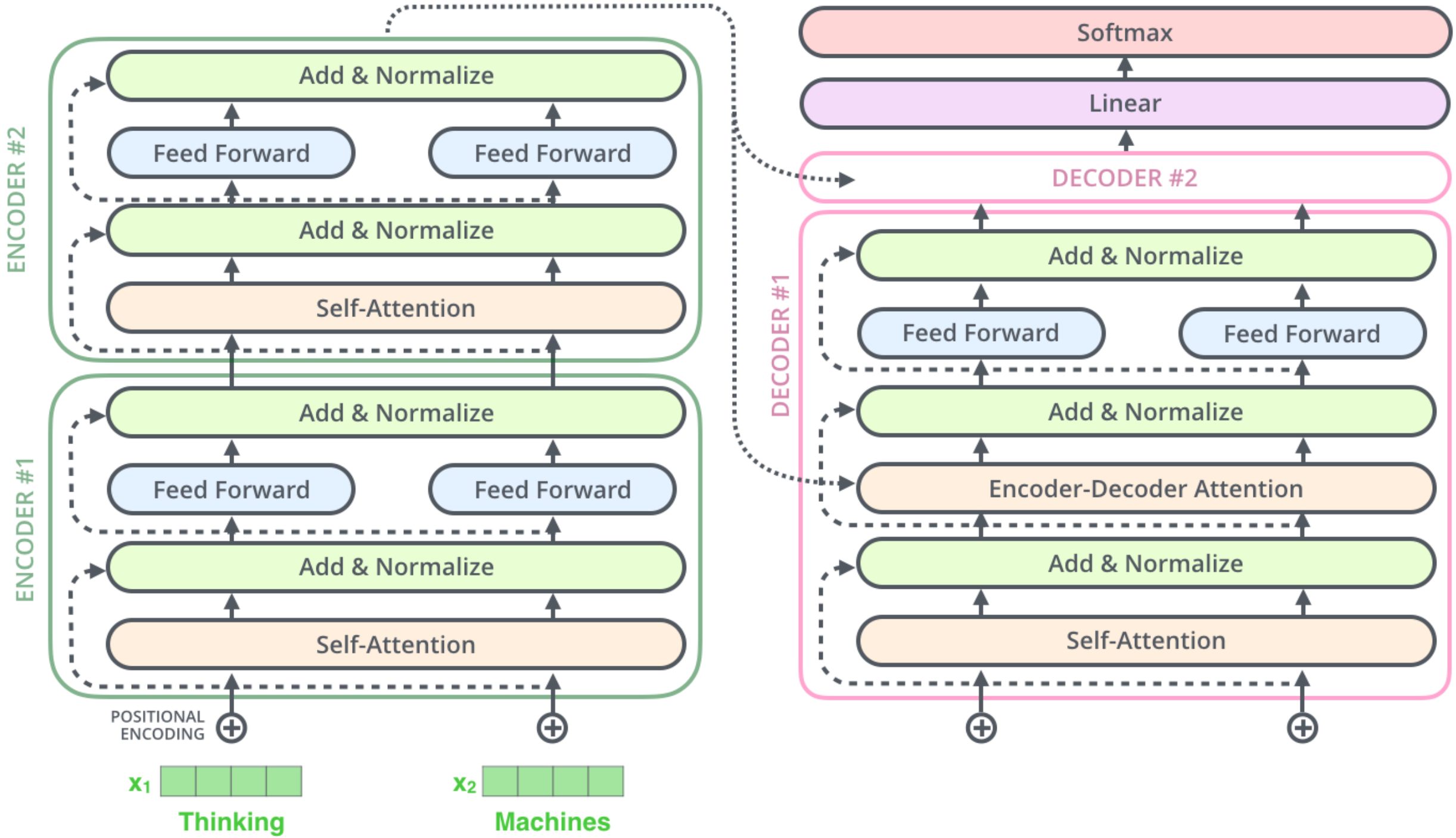
Positional encoding

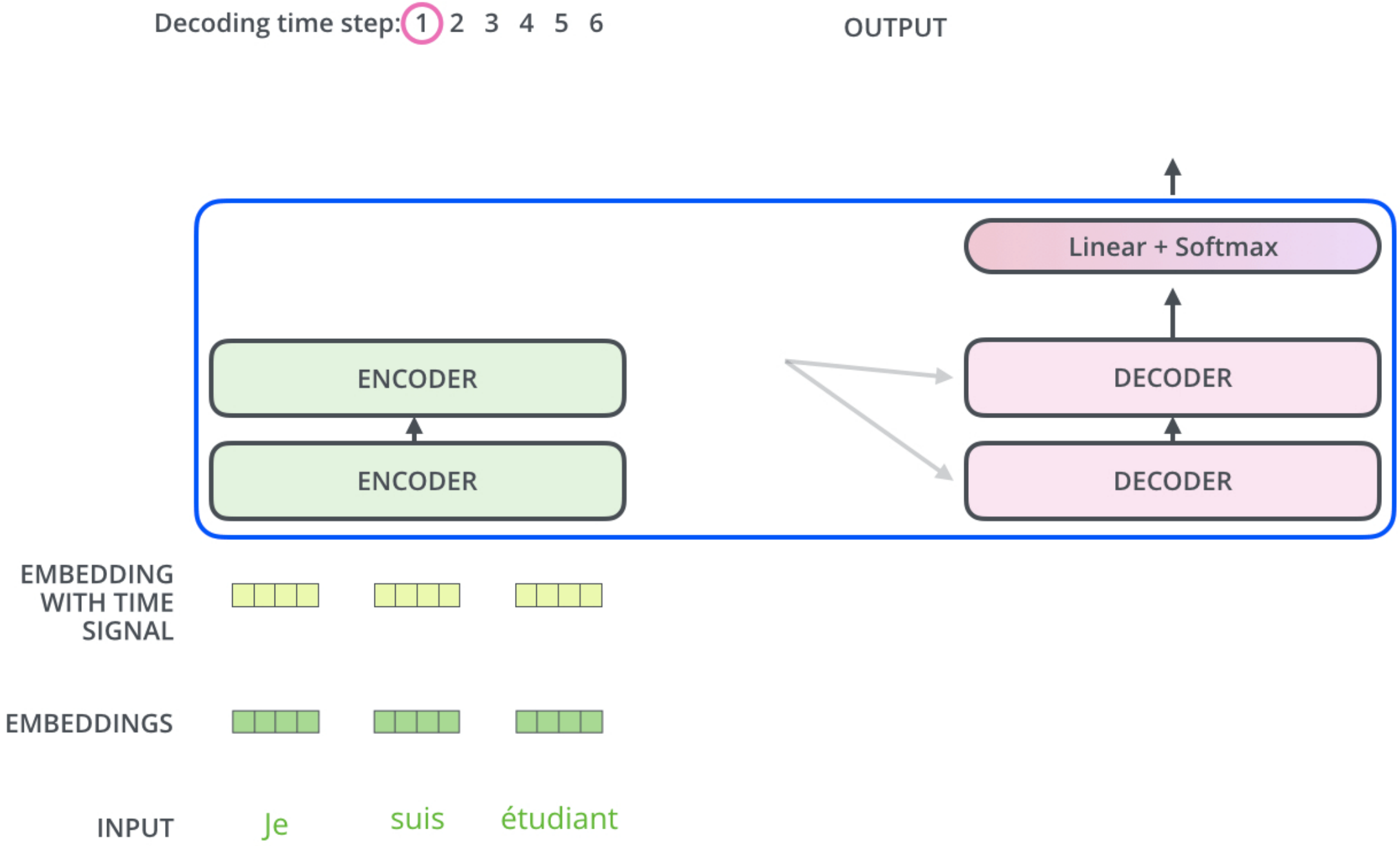


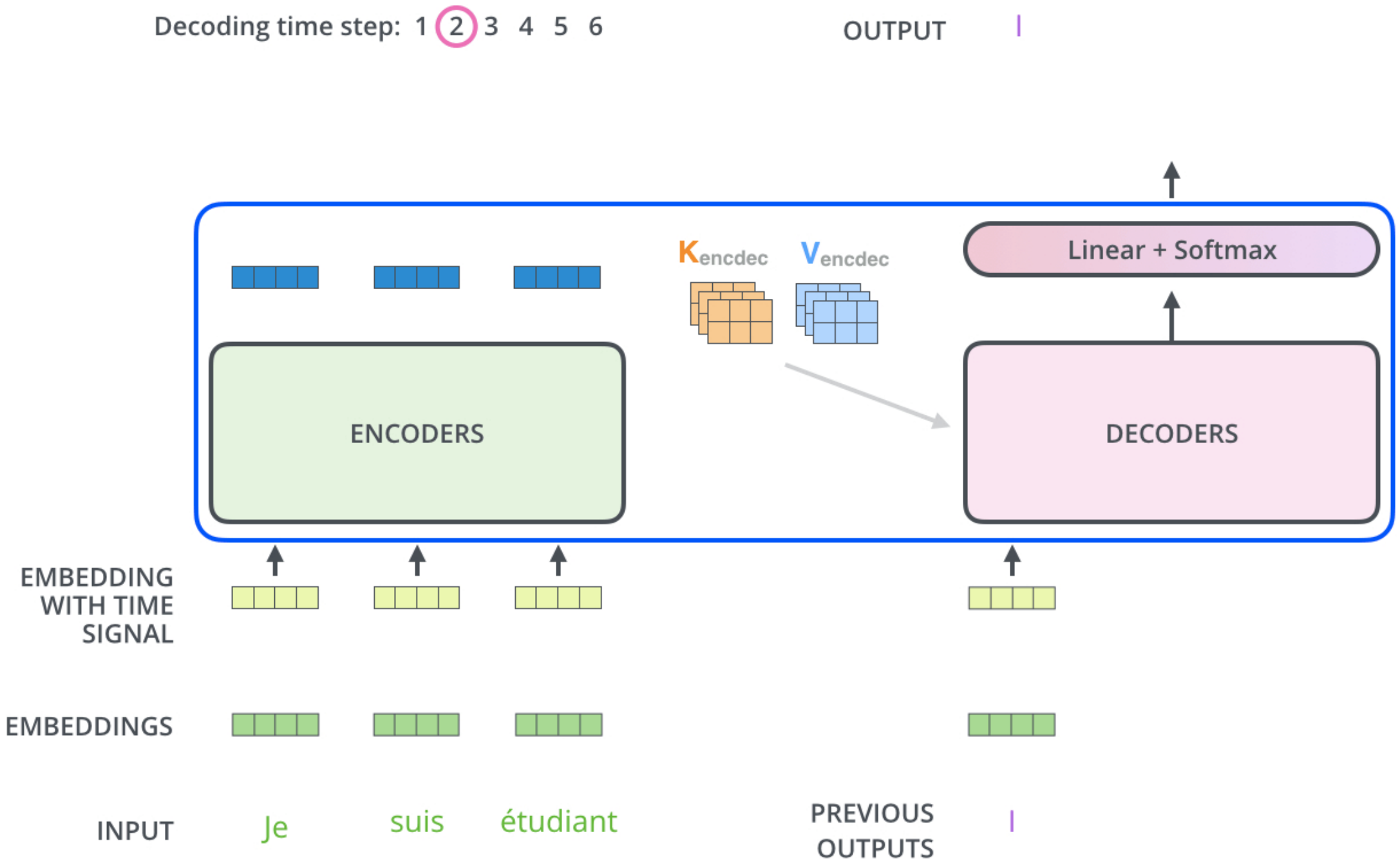
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

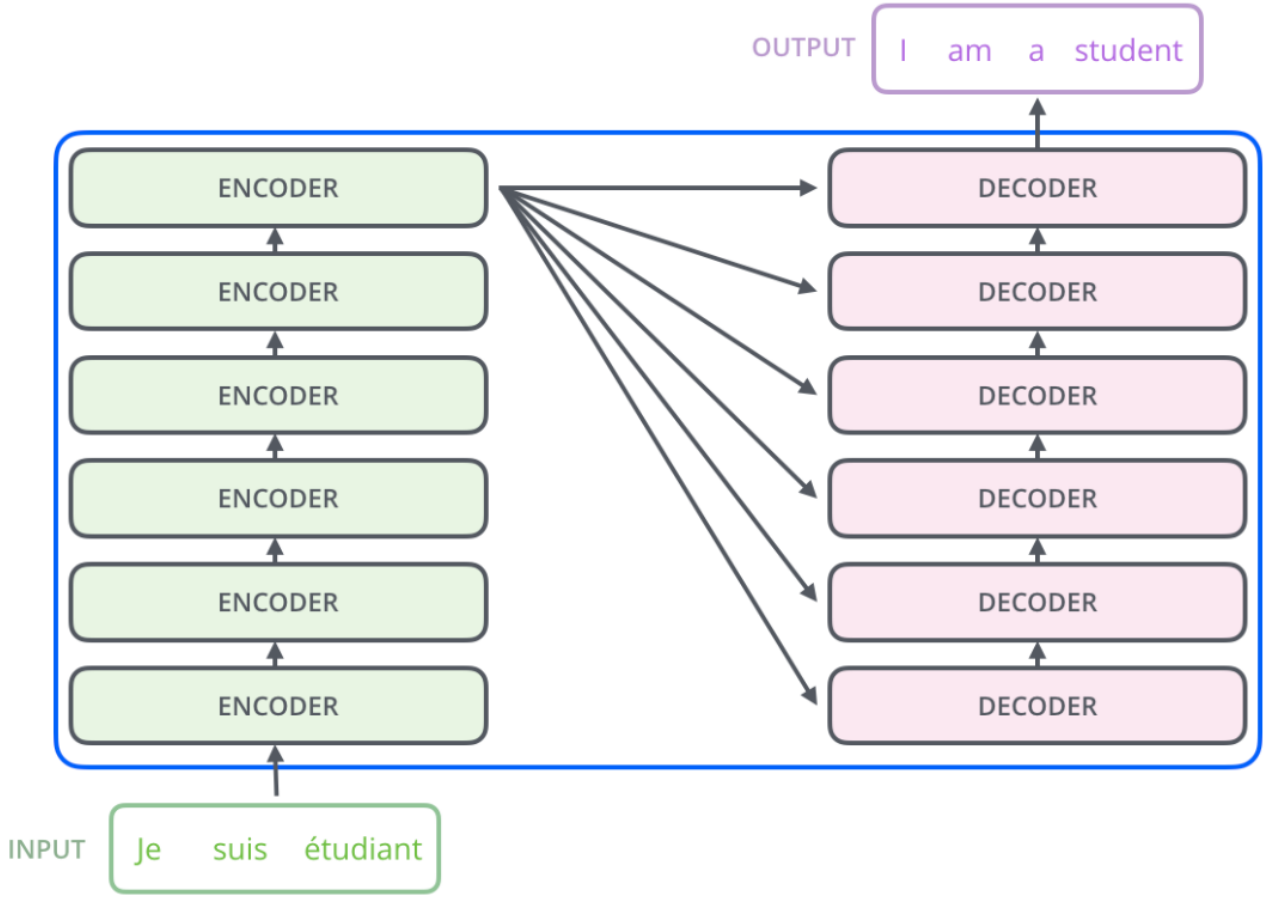
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

For any fixed offset k, $PE_{(pos+k)}$ is a linear function of $PE_{(pos)}$

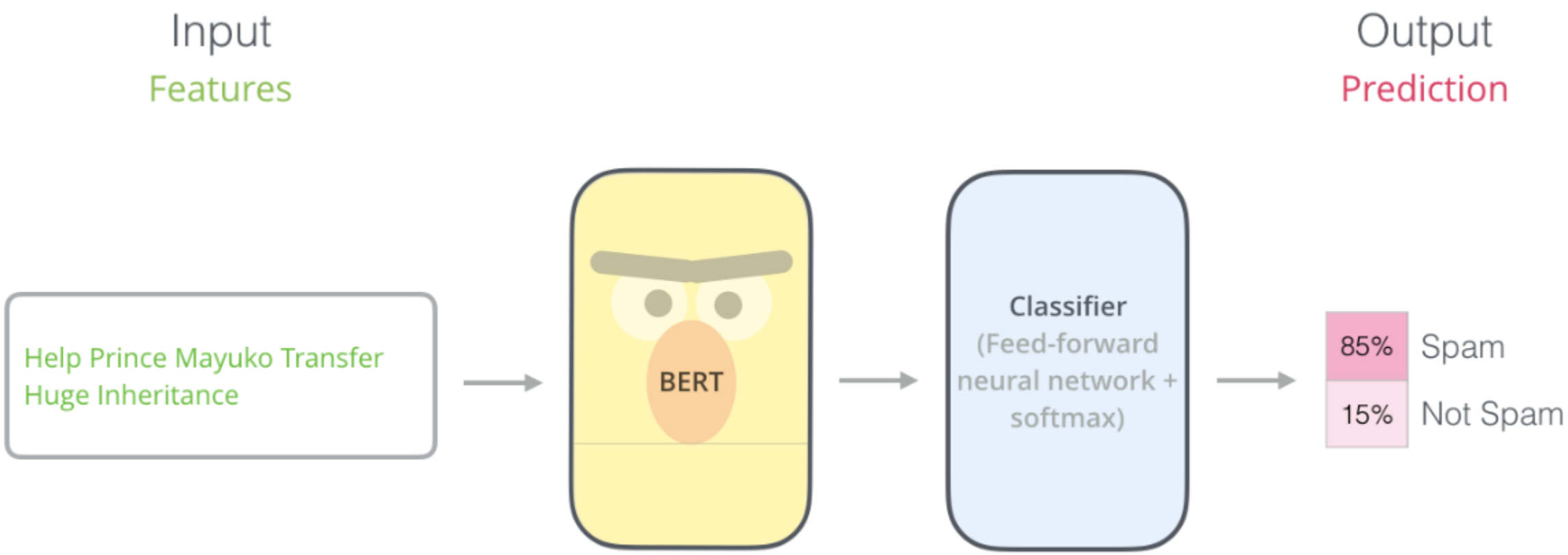

Softmax
Linear
DECODER #2
DECODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Encoder-Decoder Attention
Add & Normalize
Self-Attention
ENCODER #2
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
ENCODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
POSITIONAL ENCODING
x_1
x_2
Thinking
Machines


Decoding time step: 1 2 3 4 5 6
OUTPUT
Linear + Softmax
ENCODER
ENCODER
DECODER
DECODER
EMBEDDING WITH TIME SIGNAL
EMBEDDINGS
INPUT
Je
suis
étudiant

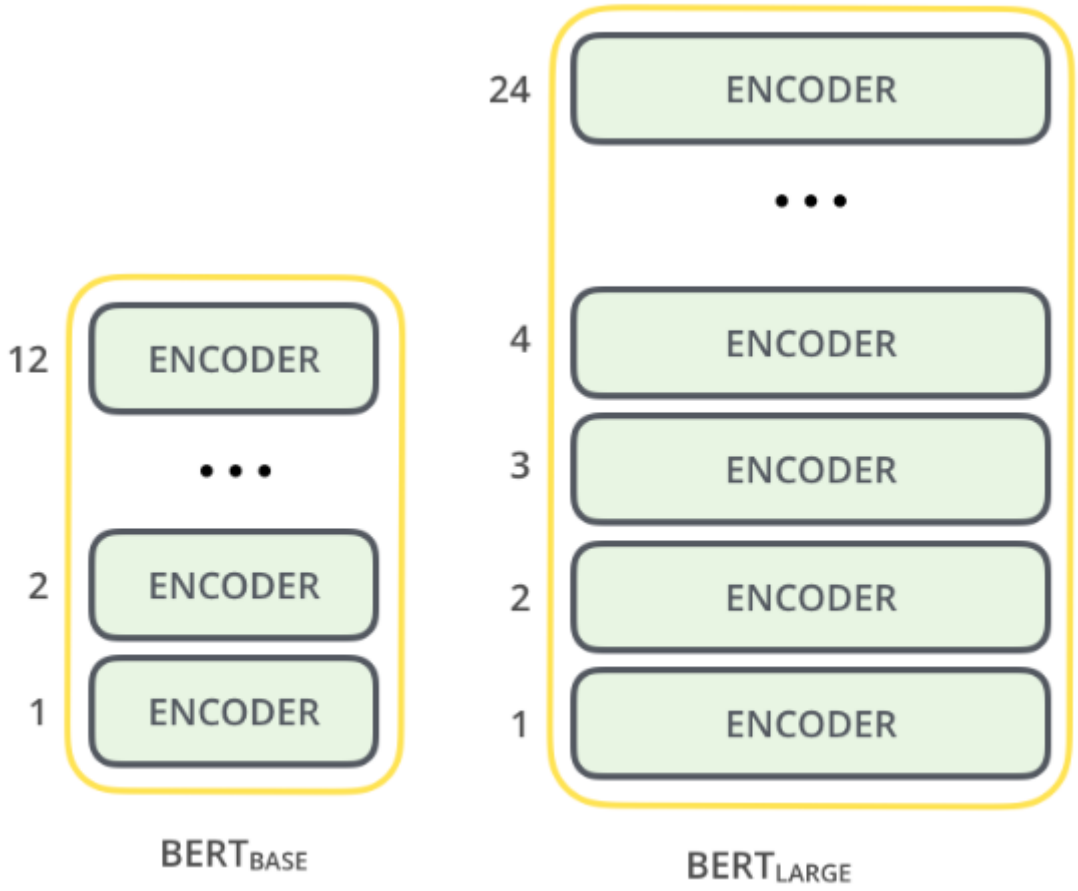

Decoding time step: 1 2 3 4 5 6
OUTPUT
I
Linear + Softmax
Kencdec
Vencdec
ENCODERS
DECODERS
EMBEDDING WITH TIME SIGNAL
EMBEDDINGS
INPUT
Je
suis
étudiant
PREVIOUS OUTPUTS
I


OUTPUT
I am a student
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
DECODER
DECODER
DECODER
DECODER
DECODER
DECODER
INPUT
Je suis étudiant

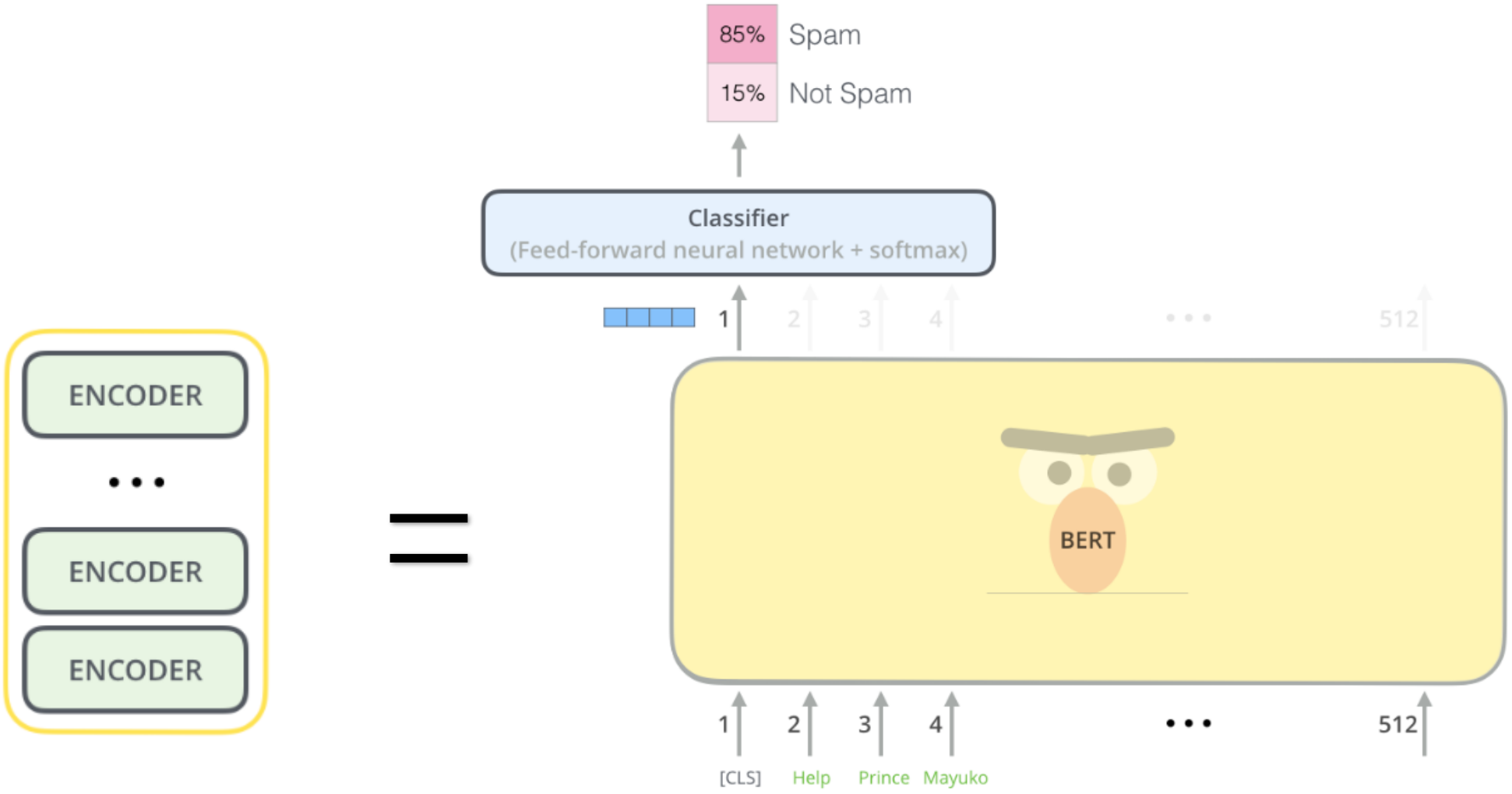
Review of BERT



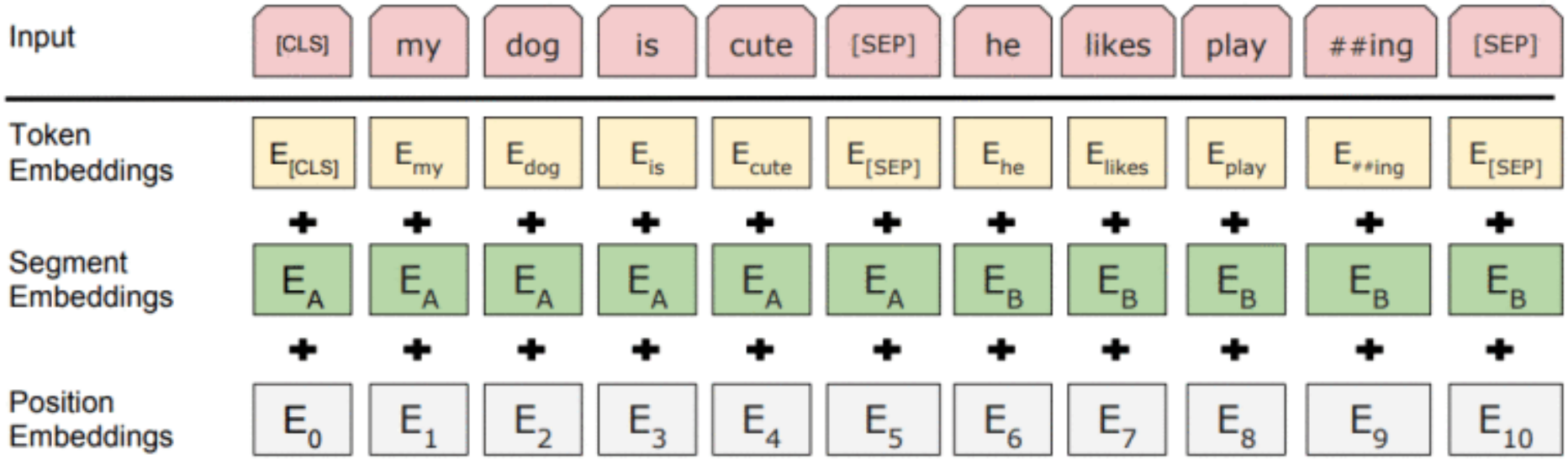
Reference: [The Illustrated BERT, ELMo, and co](#).



	Transformer	$BERT_{Base}$	$BERT_{Large}$
# of Encoders	6	12	24
# of Attention Heads	8	12	16
Hidden size	512	768	1024



[CLS]: Classification



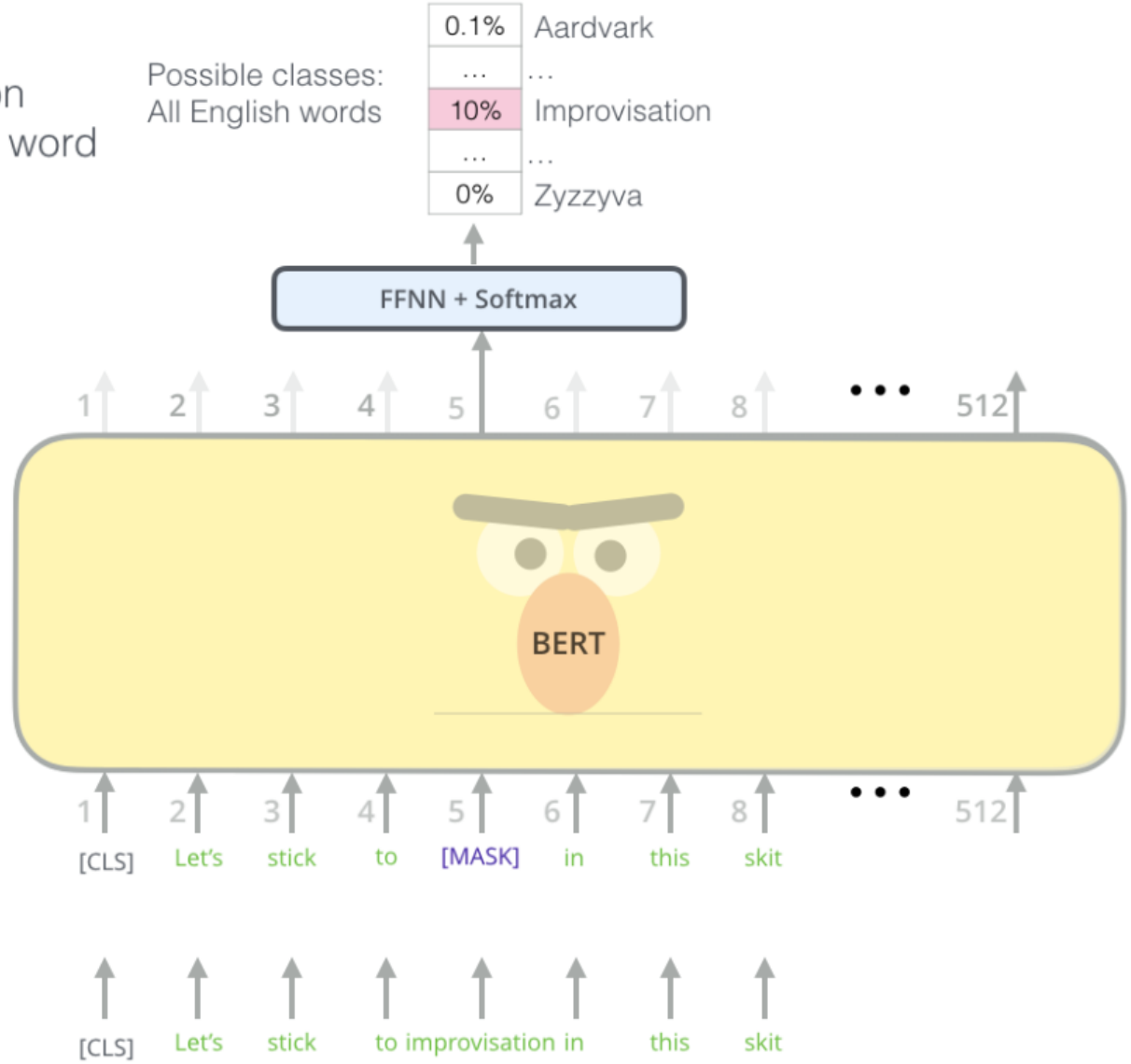
Segment: 区分不同的句子
Position: 表示位置
与Transformer不同，是学习来的

Pretraining tasks

- MLM: Masked Language Model
- NSP: Next Sentence prediction

MLM

Use the output of the masked word's position to predict the masked word

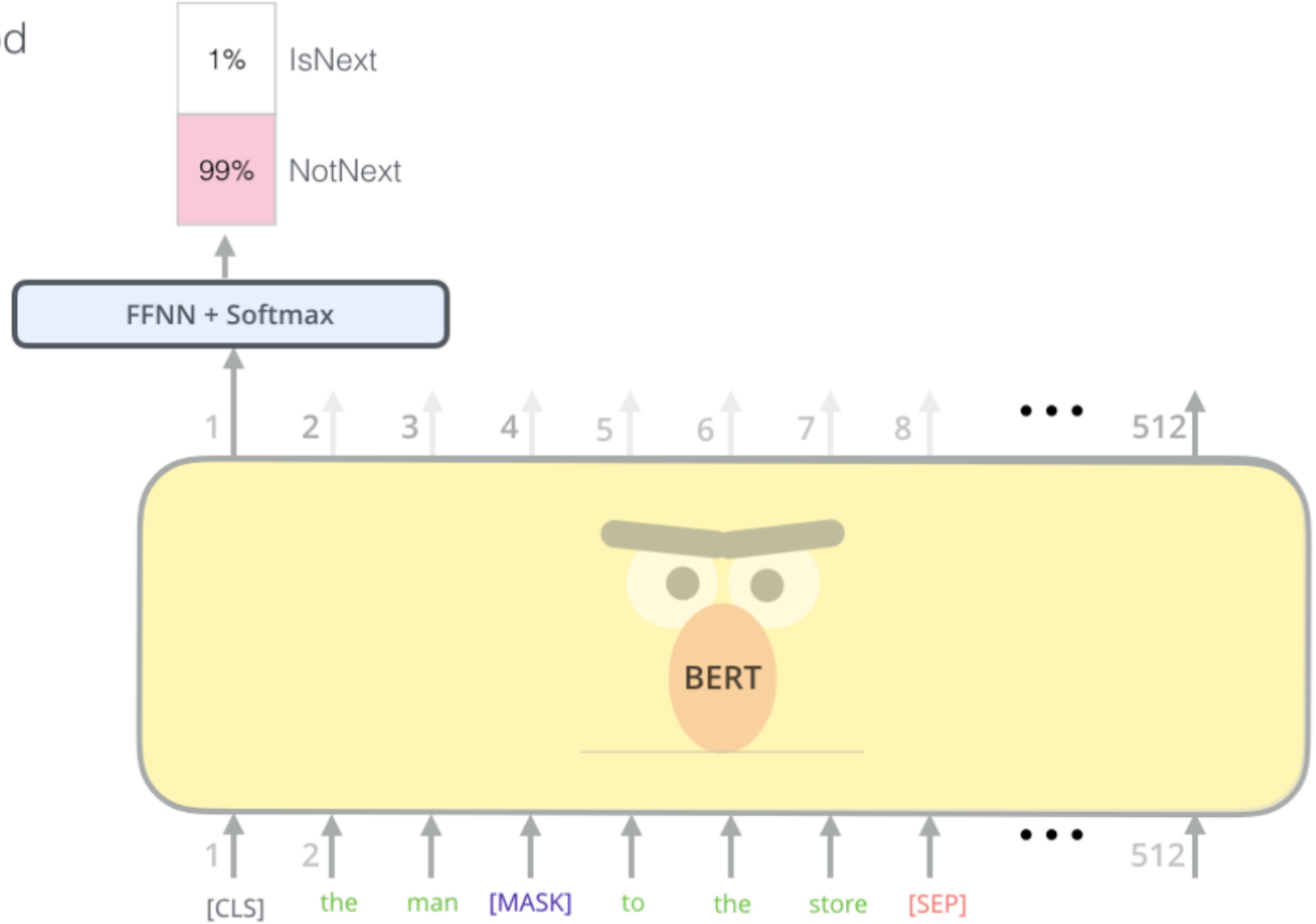


Randomly mask 15% of tokens

Input

NSP

Predict likelihood that sentence B belongs after sentence A



Tokenized Input

Input

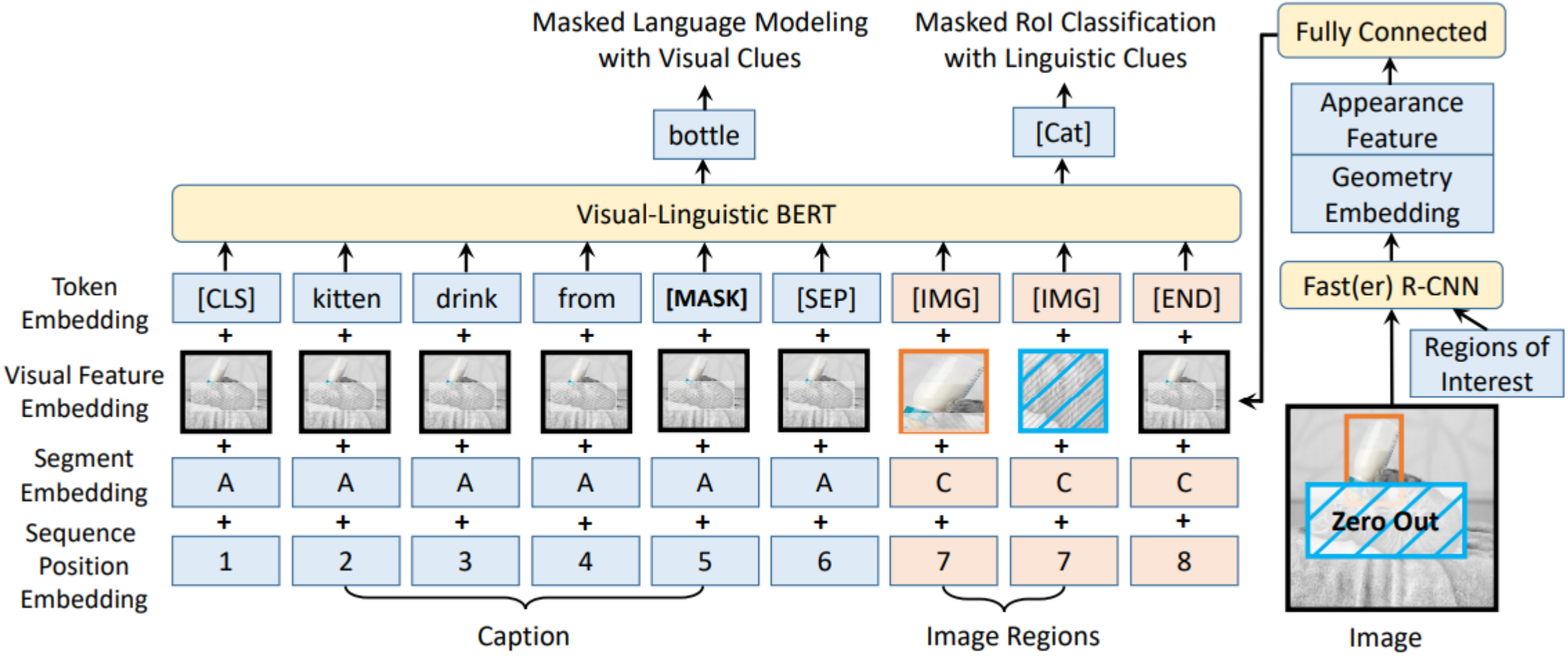
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]

Sentence A

Sentence B

VL-BERT: Pre-training of Generic Visual-Linguistic Representations

Weijie Su[1,2*], **Xizhou Zhu**[1,2*], **Yue Cao**[2], **Bin Li**[1], **Lewei Lu**[2], **Furu Wei**[2], **Jifeng Dai**[2]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
{jackroos,ezra0408}@mail.ustc.edu.cn,binli@ustc.edu.cn
{yuecao,lewlu,fuwei,jifdai}@microsoft.com



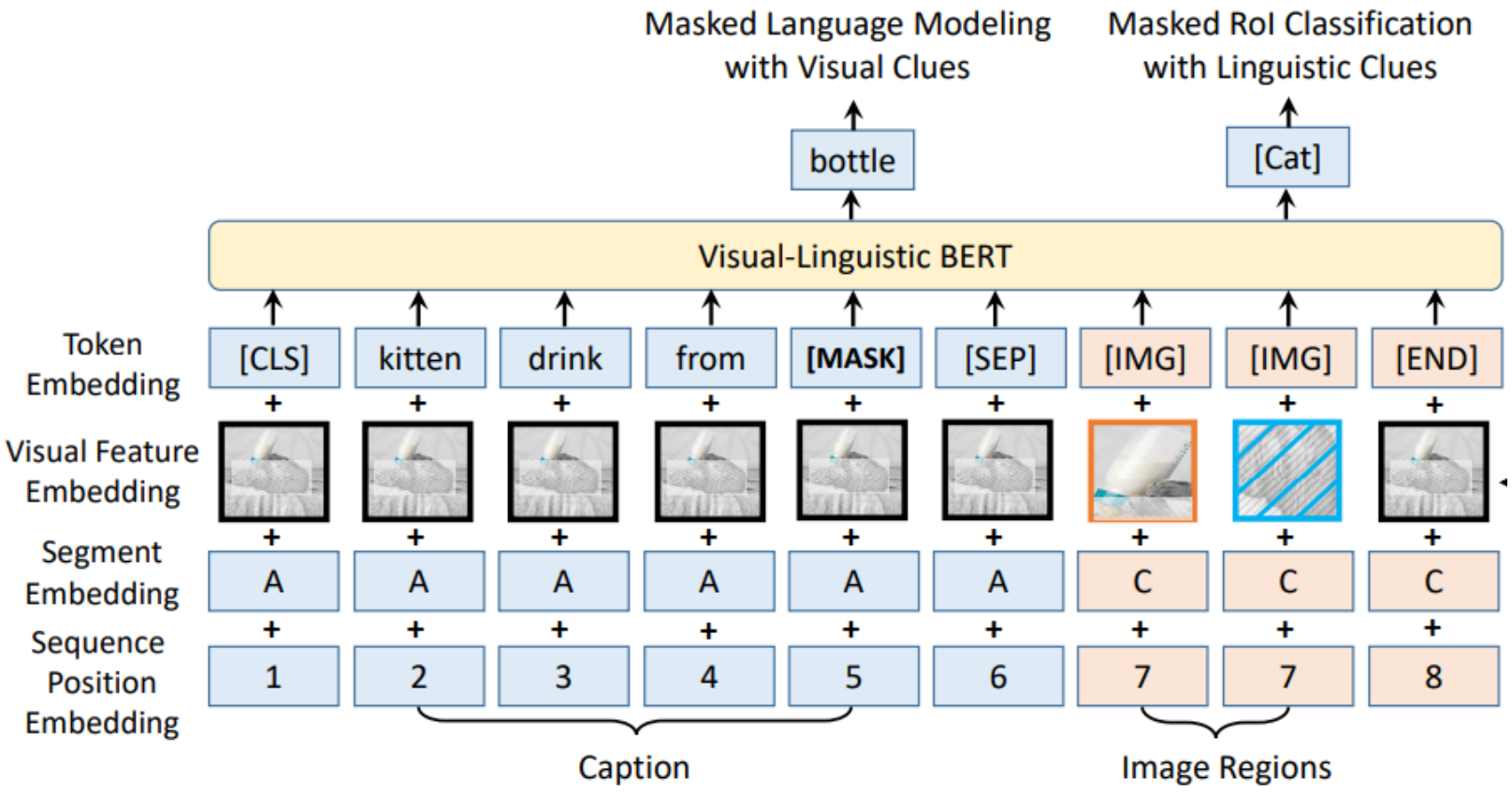
Token Embedding:
[CLS]: Classification
[SEP]: 句子分隔符
[IMG]: 图像标识
[END]: 图像结束符

Segment Embedding:
VQA: Question – A, Answer – B, Image – C
Image Caption: Caption – A, Image – C

Sequence Position Embedding:
A learnable sequence position embedding

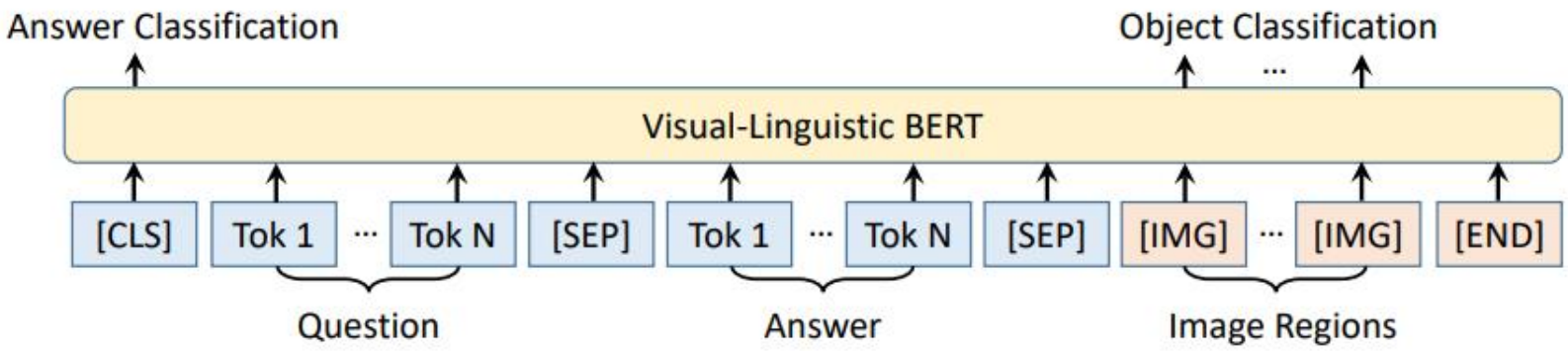
Visual Feature Embedding:
Non-visual elements: features extracted on the whole input image.
Visual elements corresponding to an RoI : features extracted by applying a Fast R-CNN detector, where the feature vector prior to the output layer of each RoI is utilized as the visual feature embedding

- Pretraining Tasks
  - Masked Language Modeling with Visual Clues
    - Each word in the input sentence(s) is randomly masked (at a probability of 15%).
  - Masked RoI Classification with Linguistic Clues
    - Each RoI in image is randomly masked out (with 15% probability).
    - the pixels laid in the masked RoI are set as zeros before applying Fast R-CNN

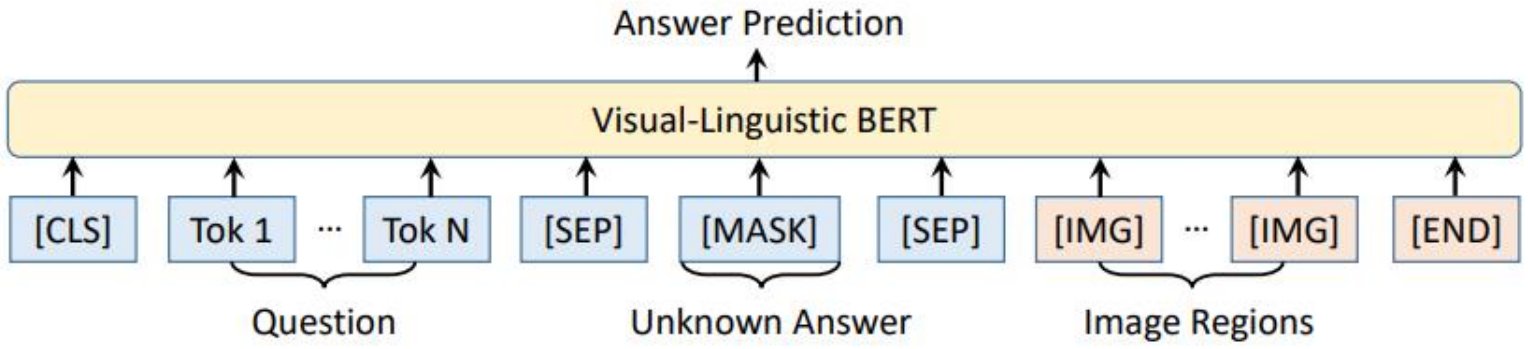


Fine-tuning

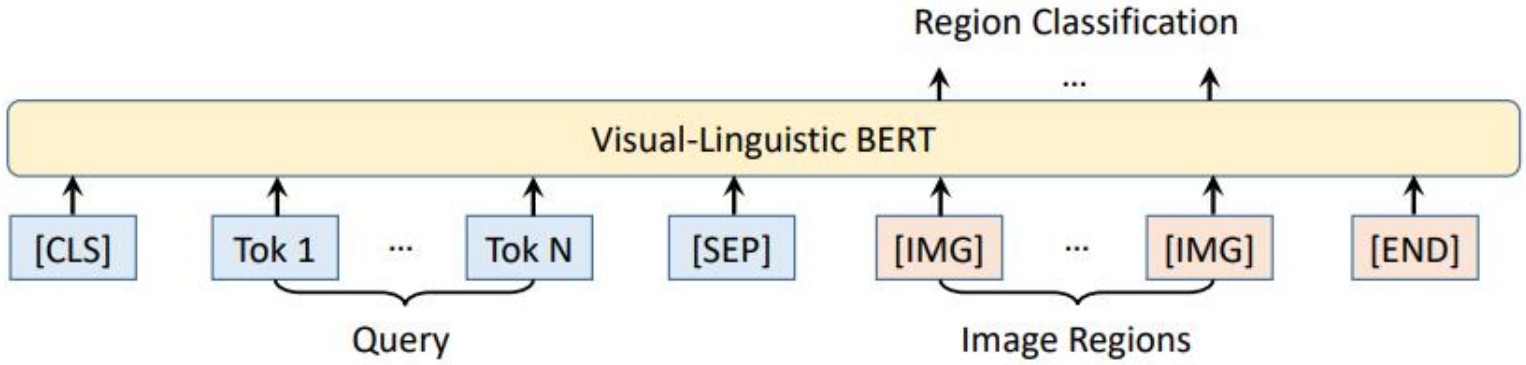
- Input:
  - <Caption, Image> and <Question, Answer, Image>
- Output:
  - [CLS] sentence-image-relation level prediction
  - Output features of words or RoIs are for word-level or RoI-level prediction



(a) Input and output format for Visual Commonsense Reasoning (VCR) dataset



(b) Input and output format for Visual Question Answering (VQA) dataset



(c) Input and output format for Referring Expression task on RefCOCO+ dataset