Transformer and its variants for NLP

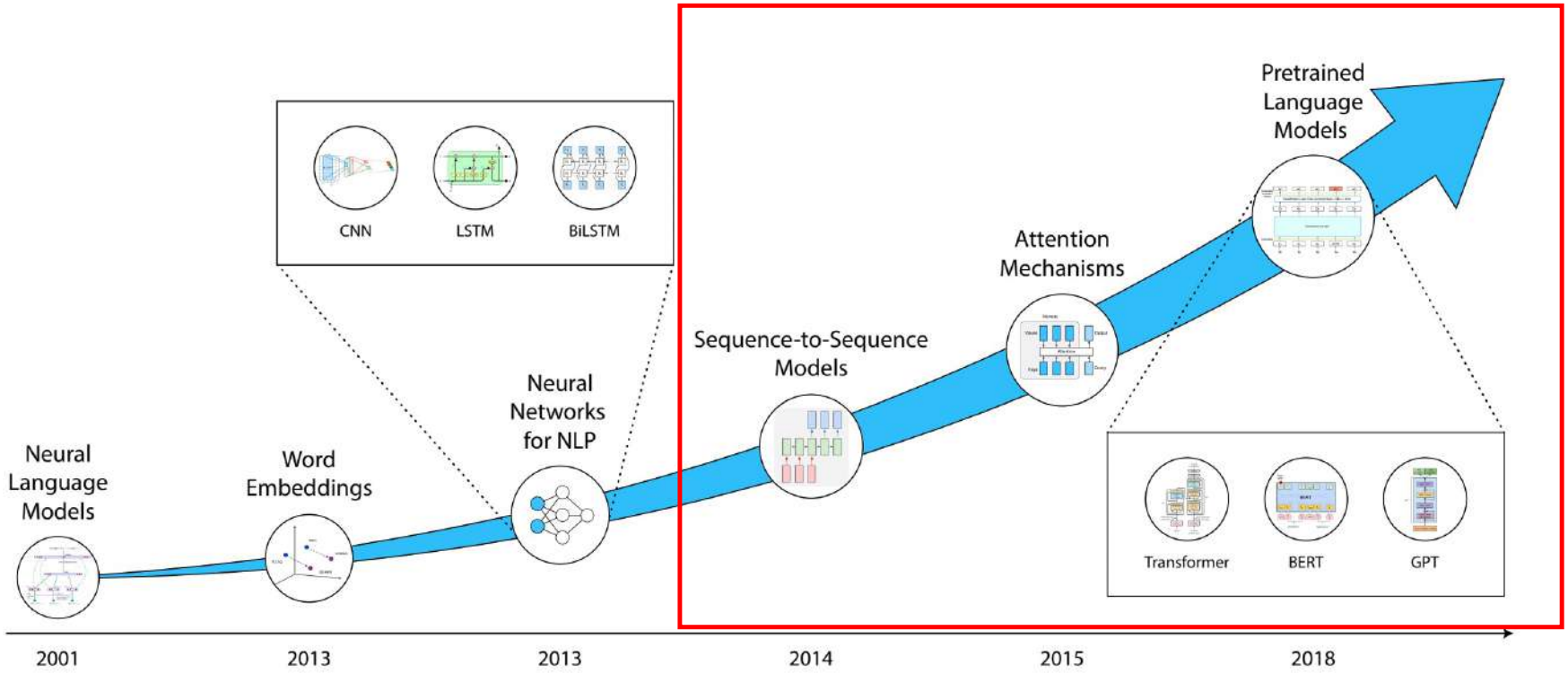
Quan Thanh Tho
qttho@hcmut.edu.vn

VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

Assoc. Prof. Quan Thanh Tho
Vice Dean
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University - Ho Chi Minh City
qttho@hcmut.edu.vn
http://www.cse.hcmut.edu.vn/qttho/

- BEng, HCMUT, Vietnam, 1998
- PhD, NTU, Singapore, 2006
- Research Interests: Artificial Intelligence, Natural Language Processing, intelligent systems, formal methods

NLP Milestones



Quan Thanh Tho, "Modern Approaches in Natural Language Processing", *VNU Journal of Science: Computer Science and Communication Engineering*, 2022

Agenda

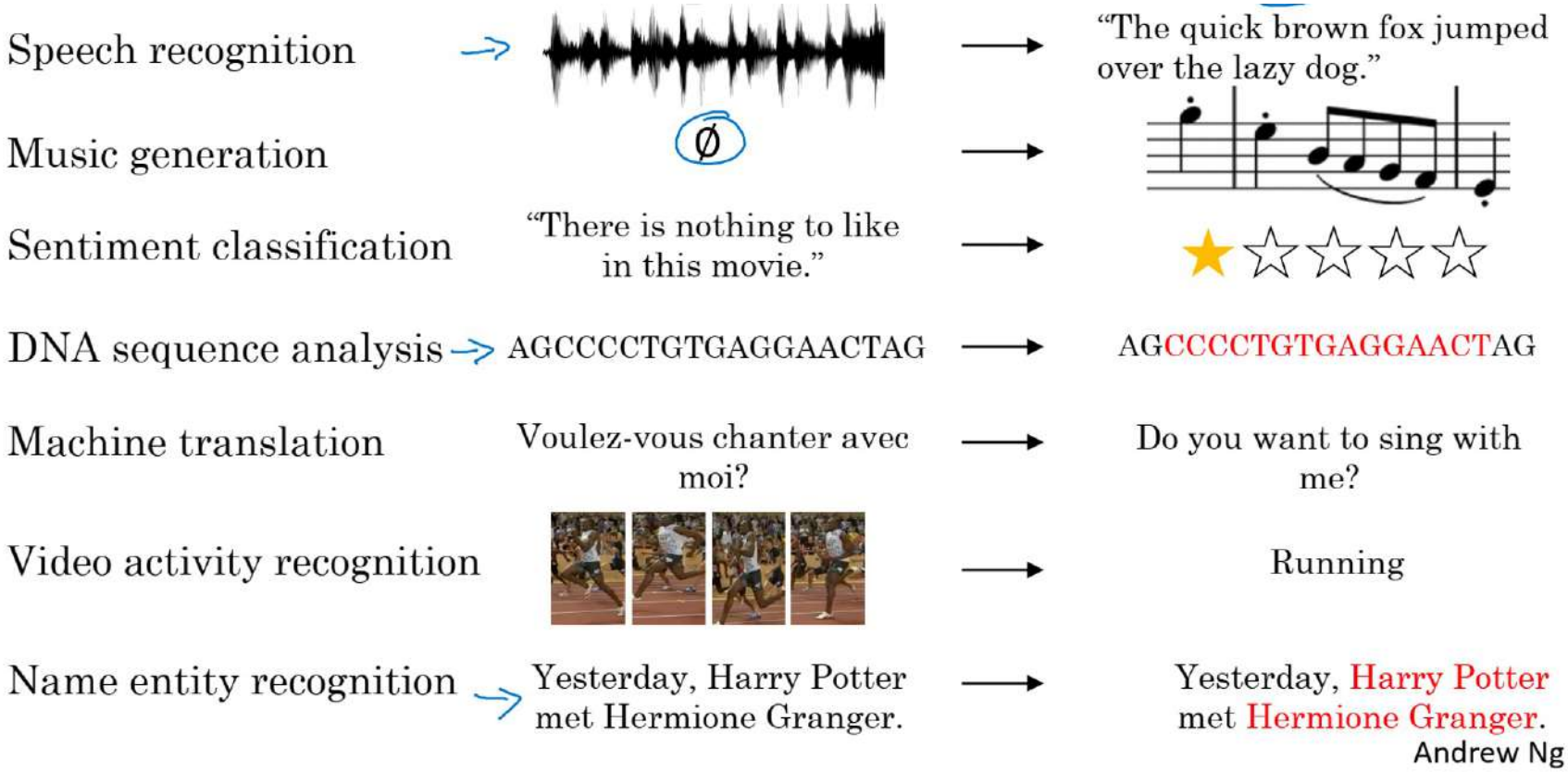
- Sequence data and sequence models
- Seq2Seq and attention
- Transformer model
- BERT and other variants
- Applications in NLP

Sequence data and sequence models

Sequence data

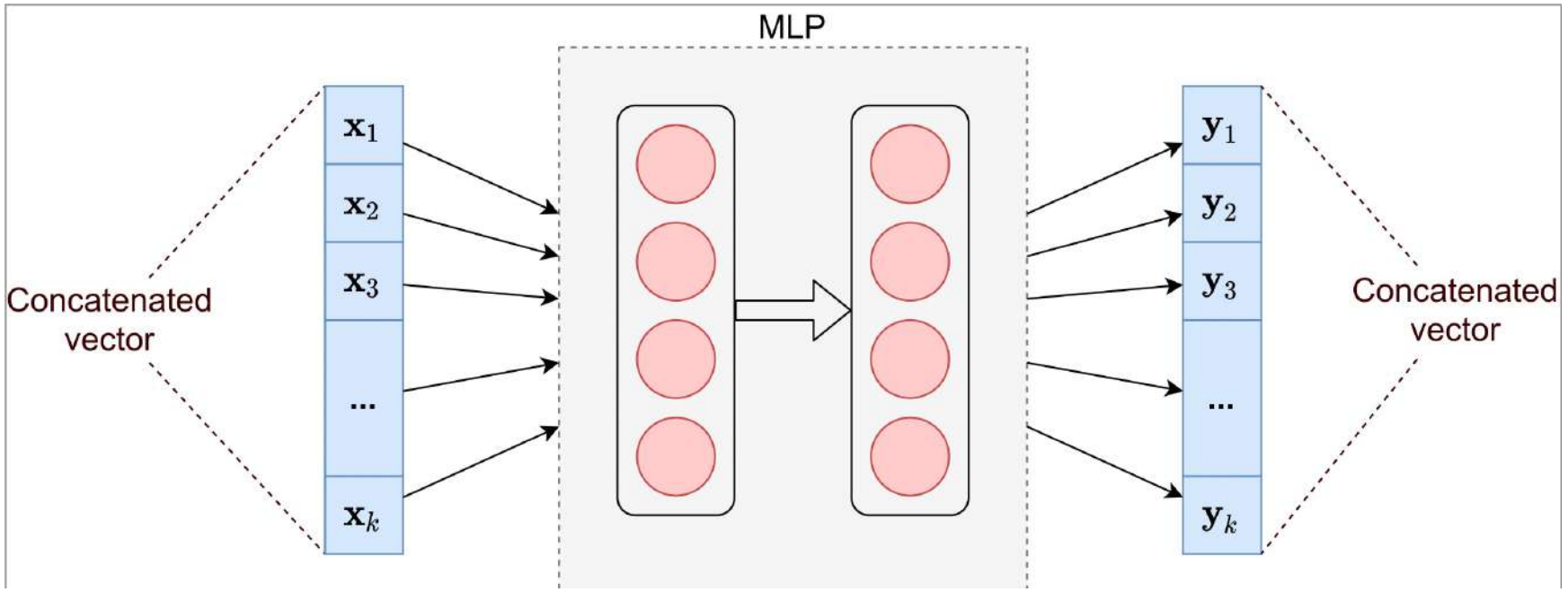
A series of data points whose points reliant on each other

- Length can be varied
- Positions matter



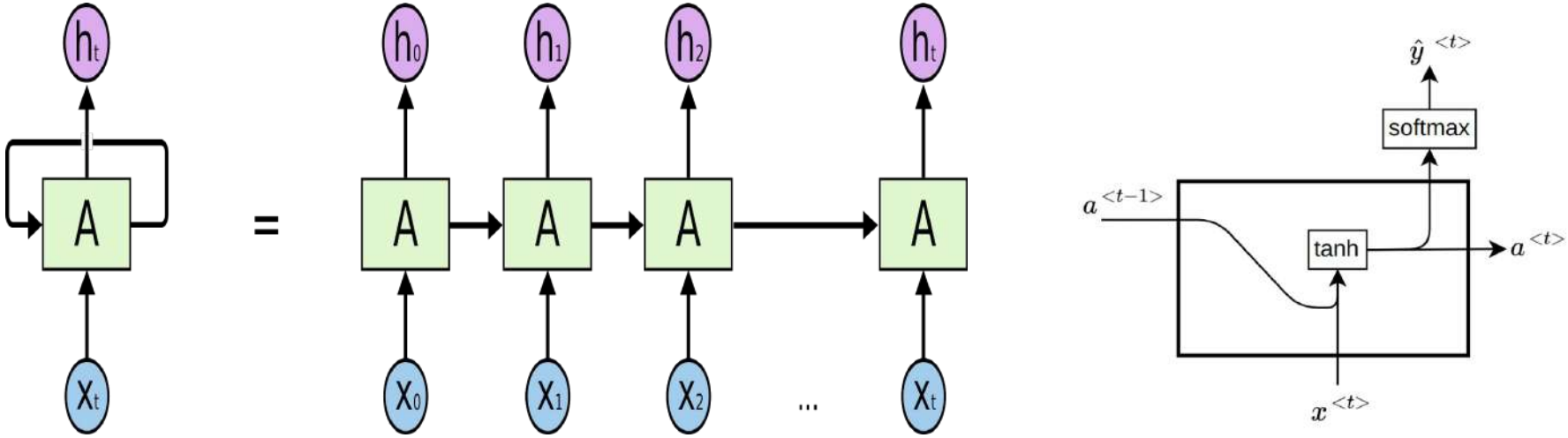
Andrew Ng

Problem of Standard Networks



- Inputs, outputs can be different lengths in different examples.
- Relations between positions are not well reflected

RNN comes as a rescue



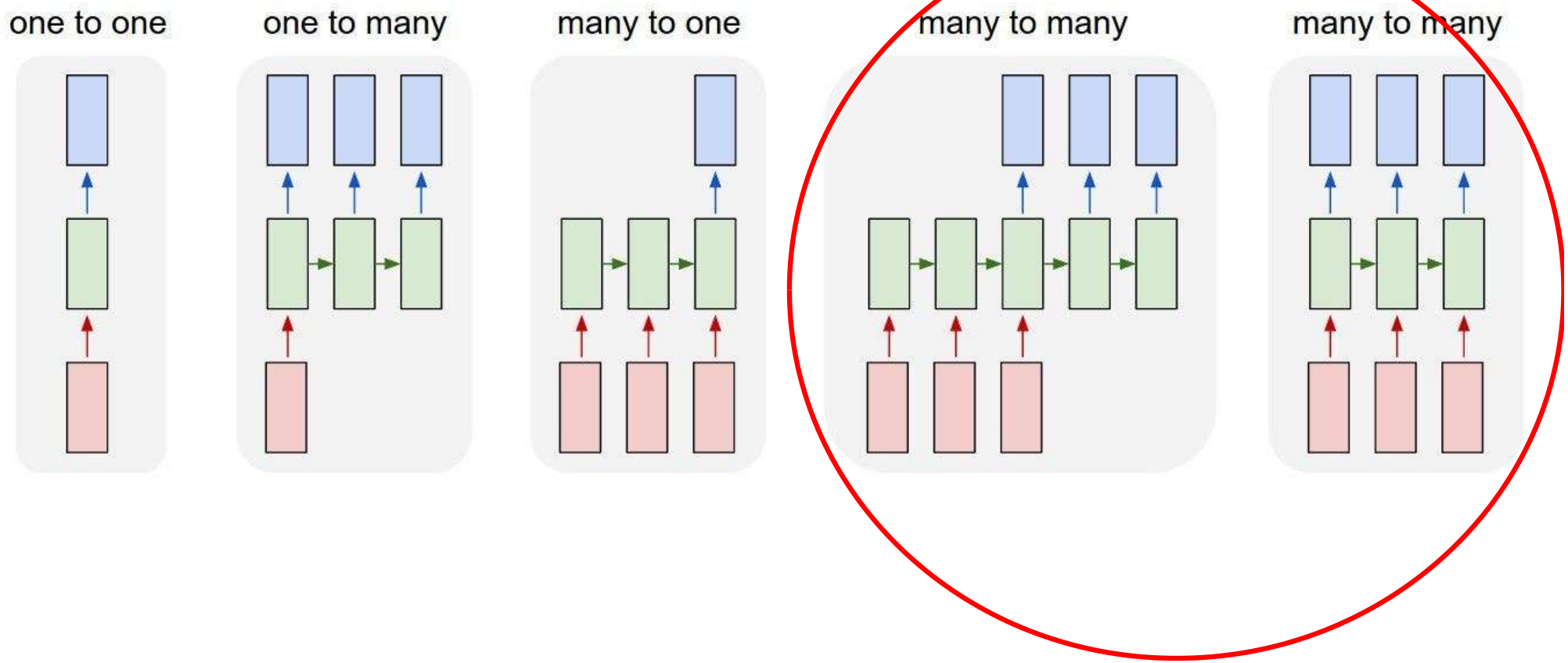
RNN: an architecture tailored for sequence data:

1) Doesn't depend on data length

2) Take advantage of past information

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. Mcclelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations* (pp. 318–362). MIT Press

RNN Revision



Seq2seq and Attention

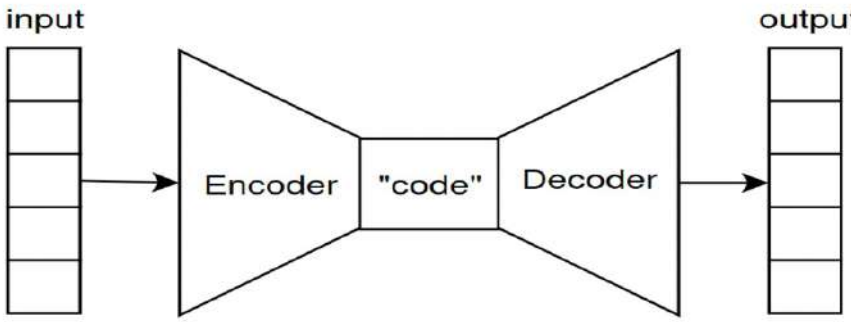
Intuition

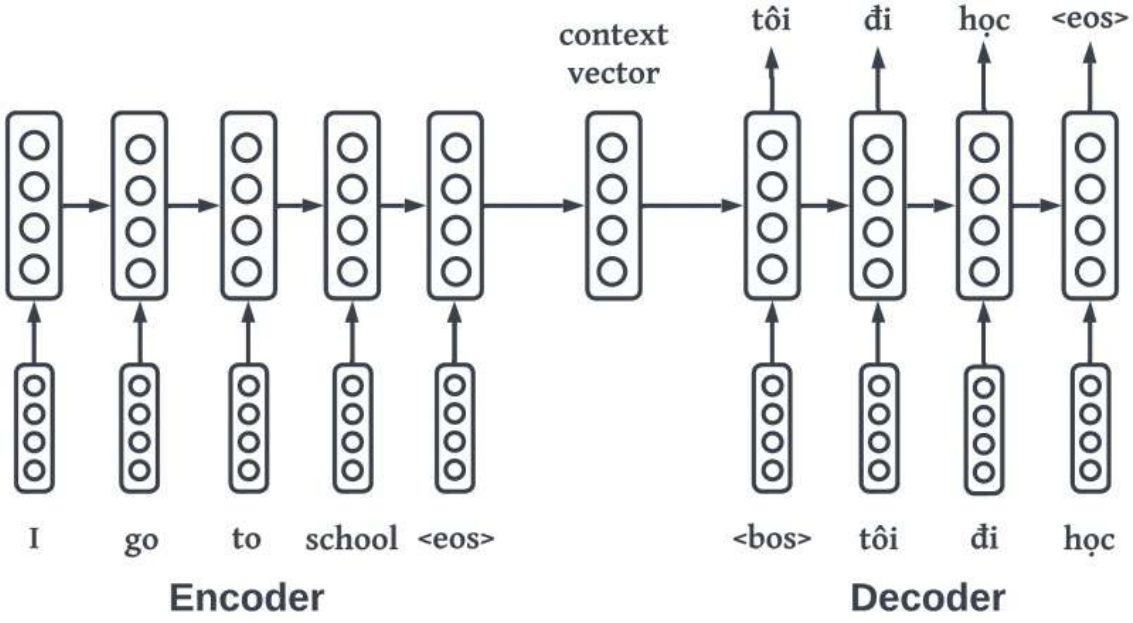
Take machine translation task as an example: human would first read some parts of the text and then start to do the translation

The cat likes to eat pizza

el gato le gusta comer pizza

Seq2Seq architecture



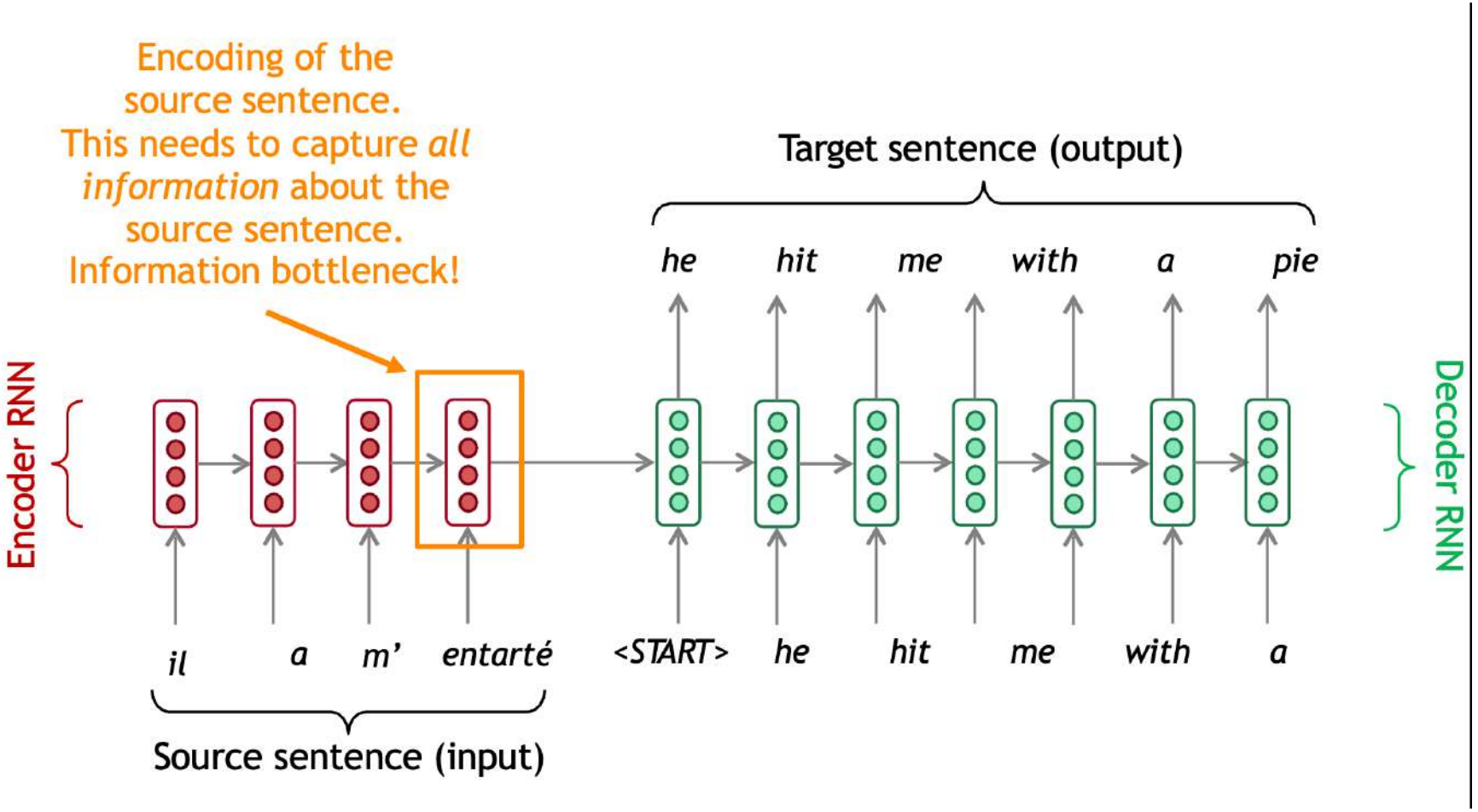


NLP researchers also employ that **idea** into designing a structure dubbed as Sequence-to-Sequence (Seq2Seq), which extends AutoEncoder architecture

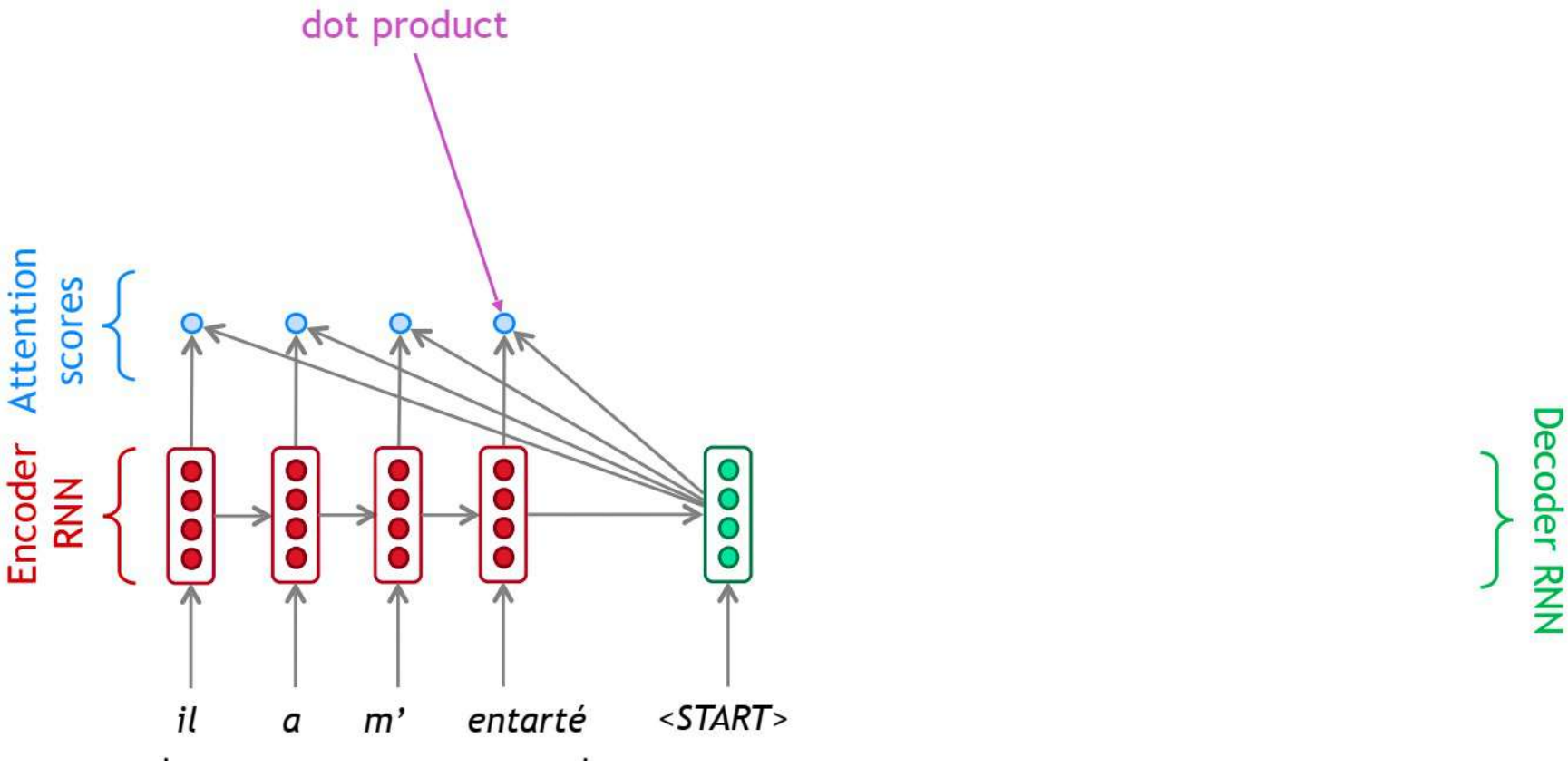
Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27

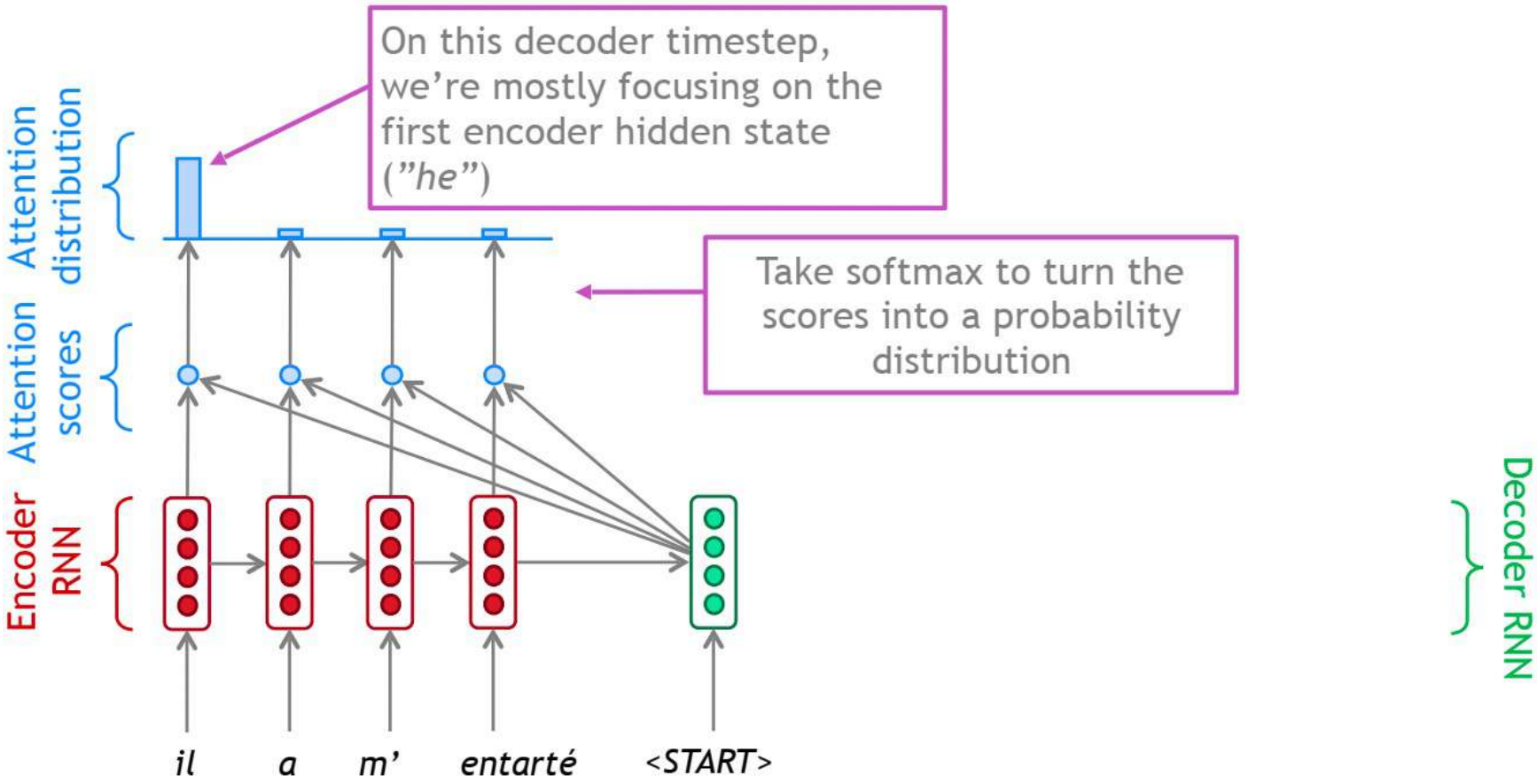
Seq2Seq: The bottle neck problem



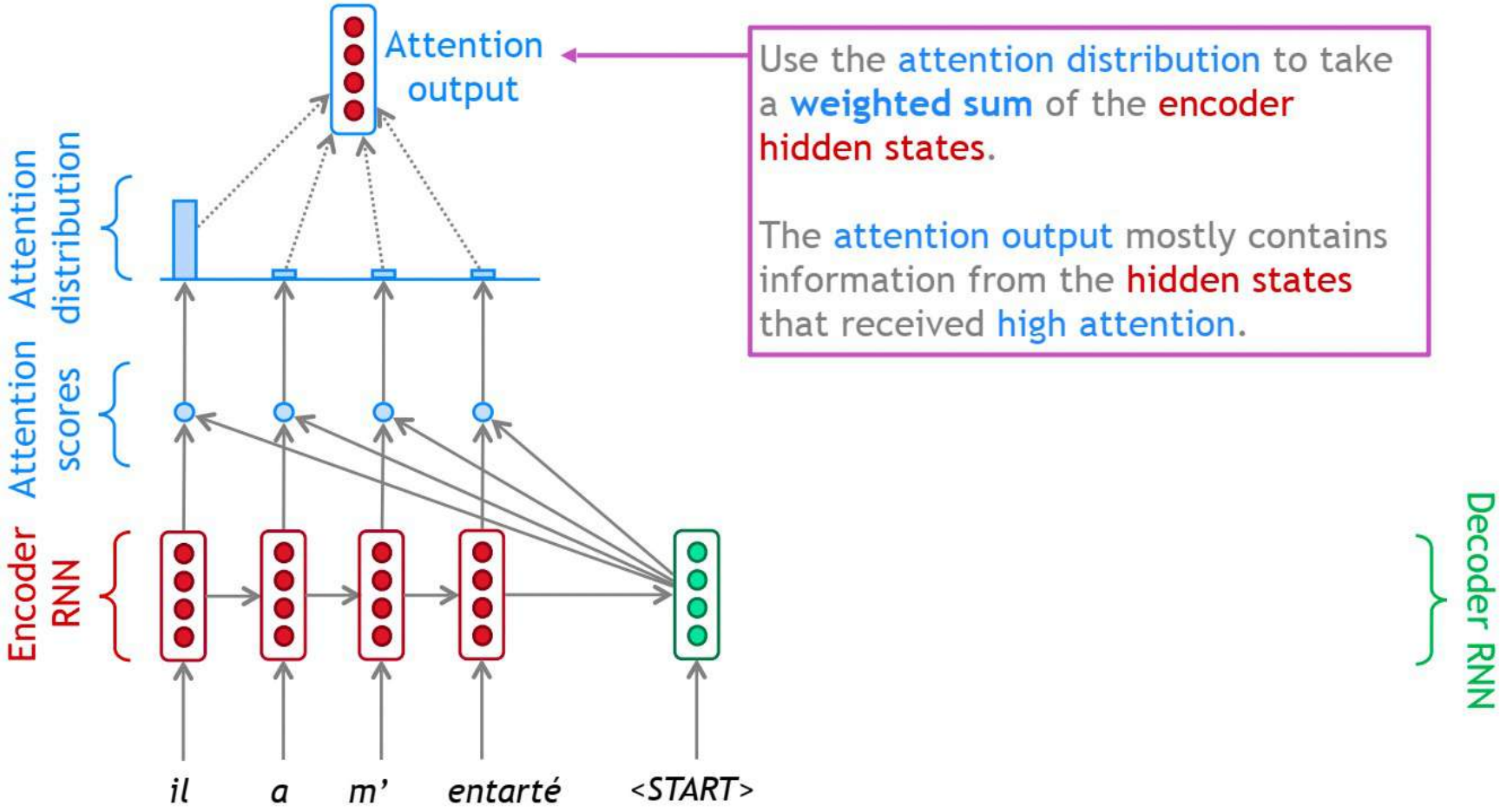
Seq2Seq with attention



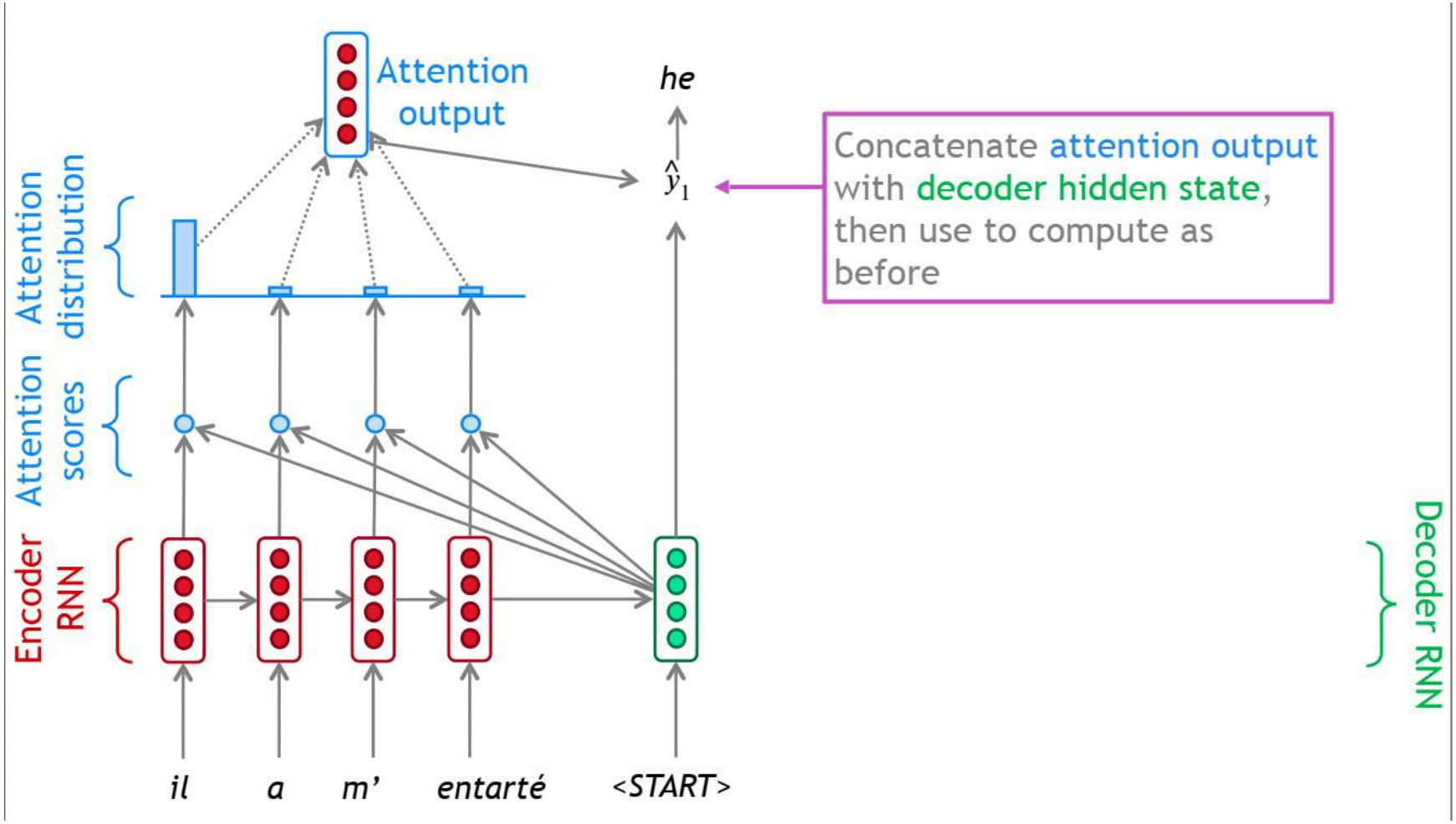
Seq2Seq with attention



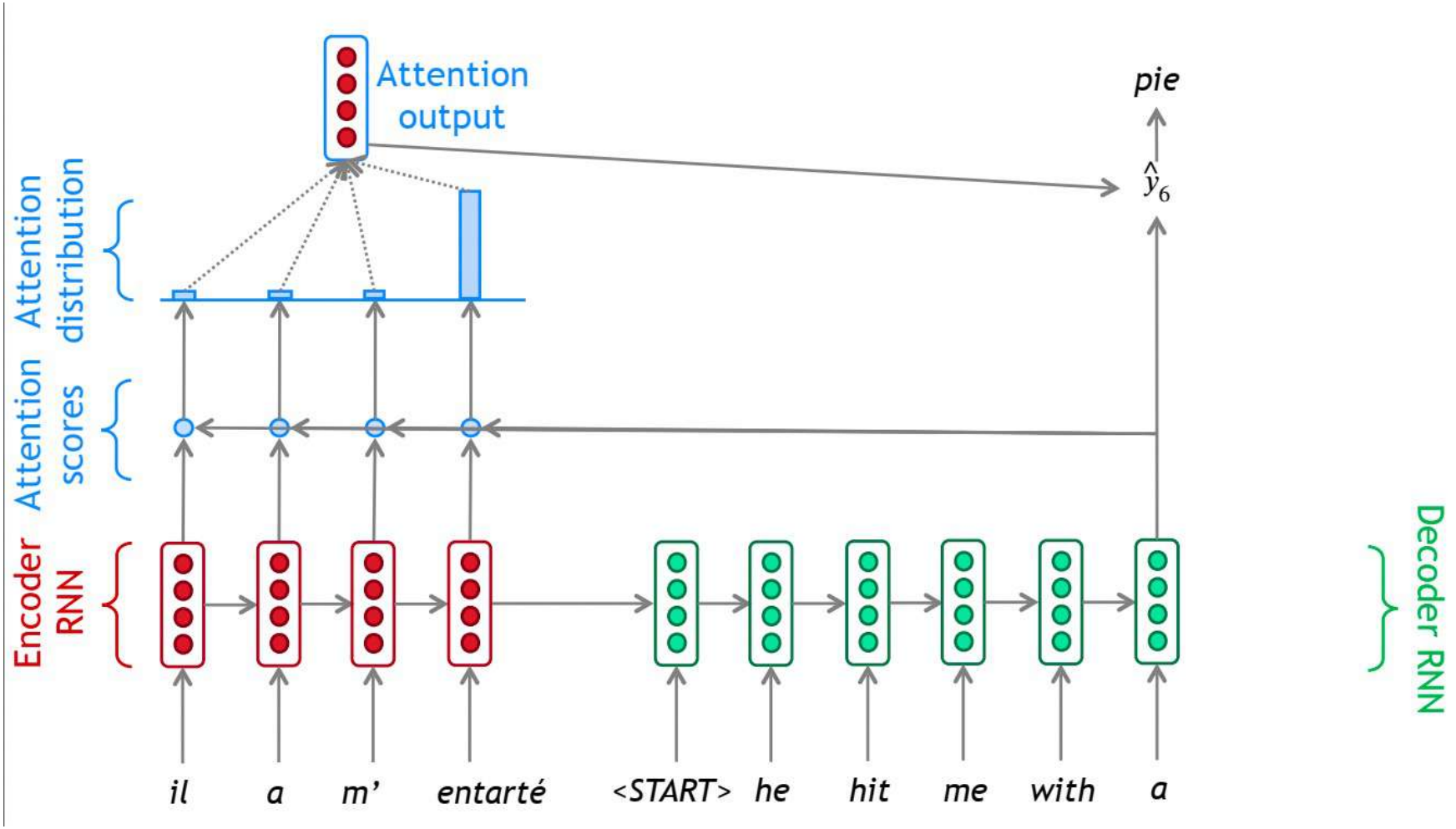
Seq2Seq with attention



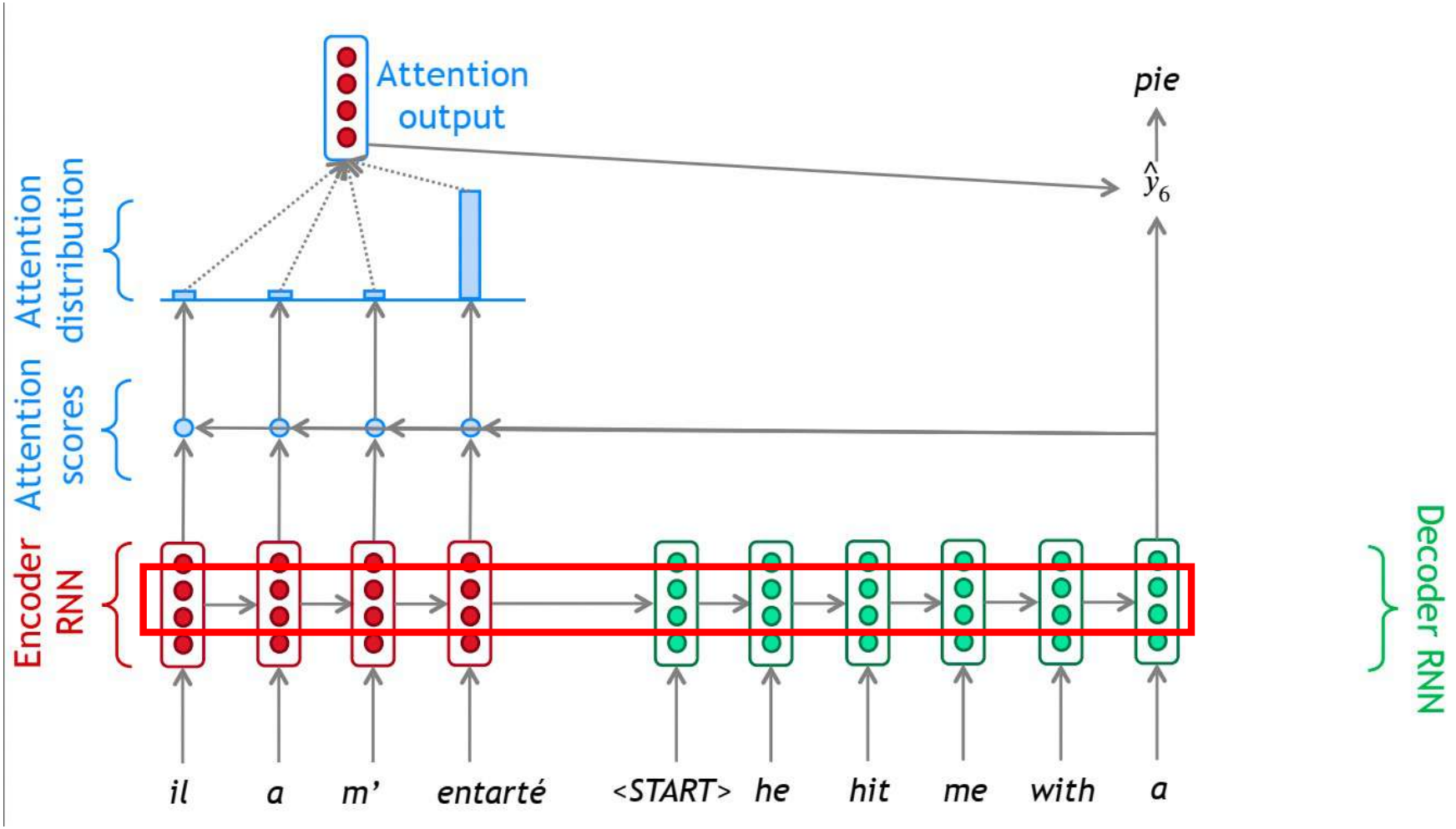
Seq2Seq with attention

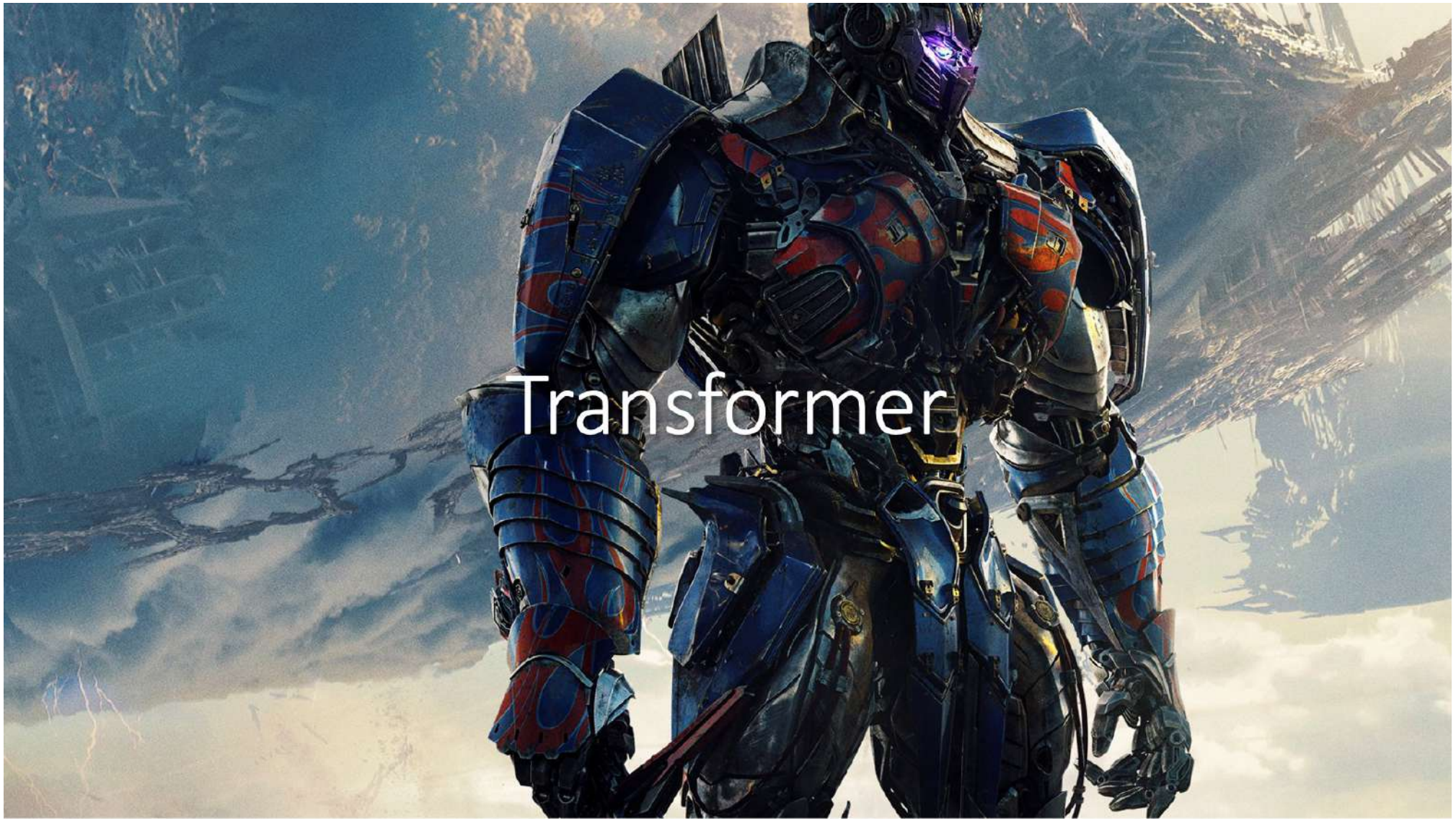


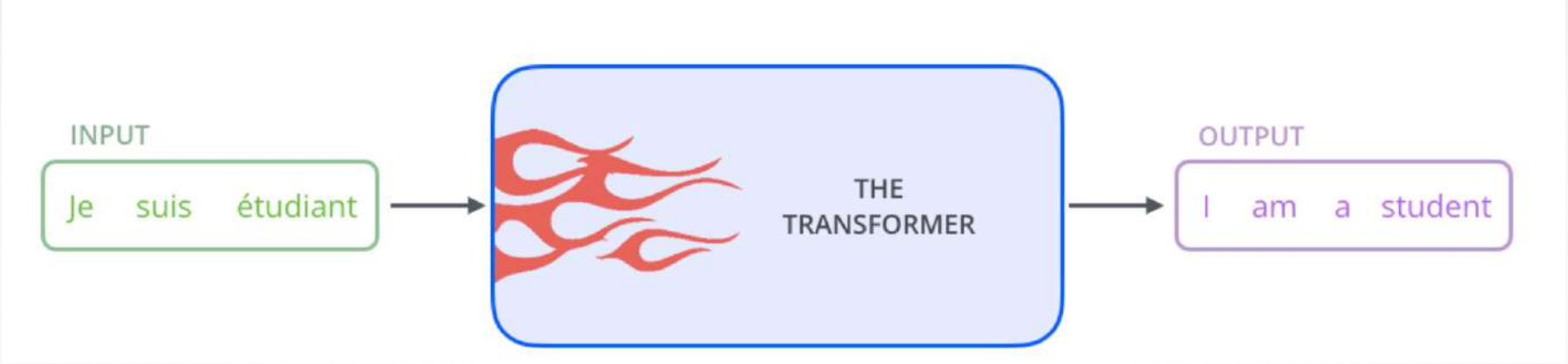
Seq2Seq with attention

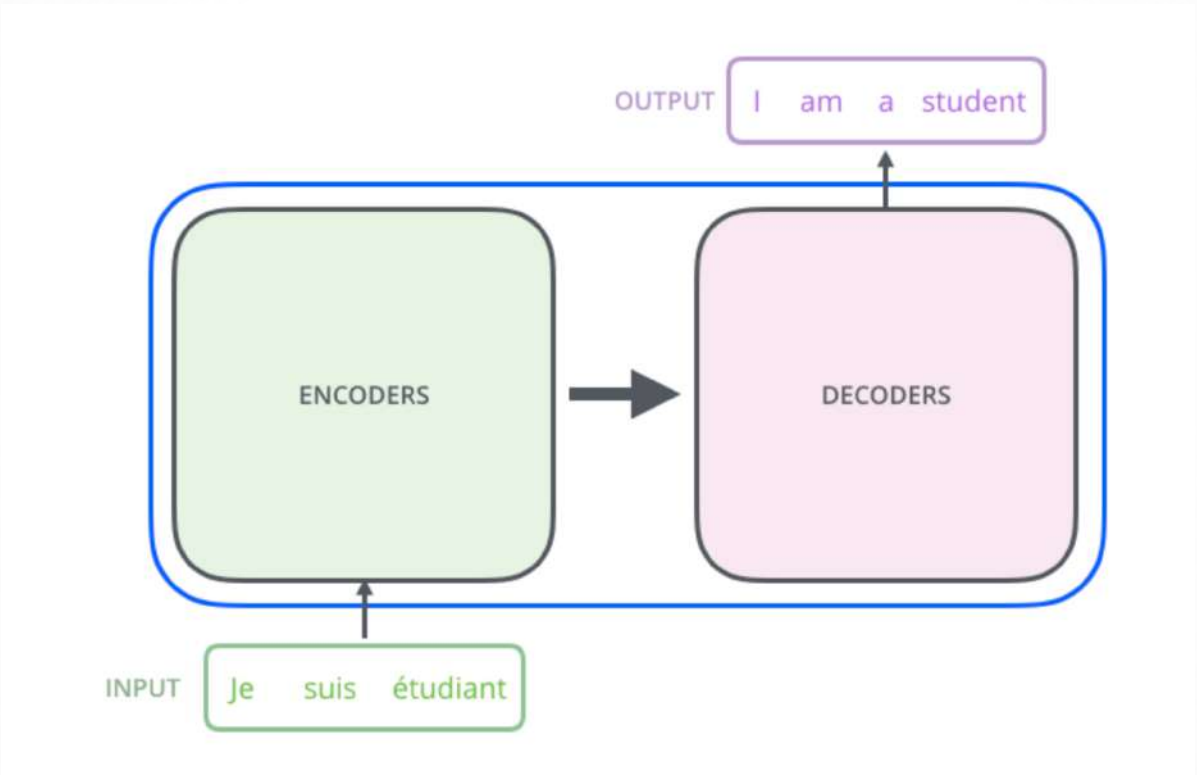


Seq2Seq with another bottleneck

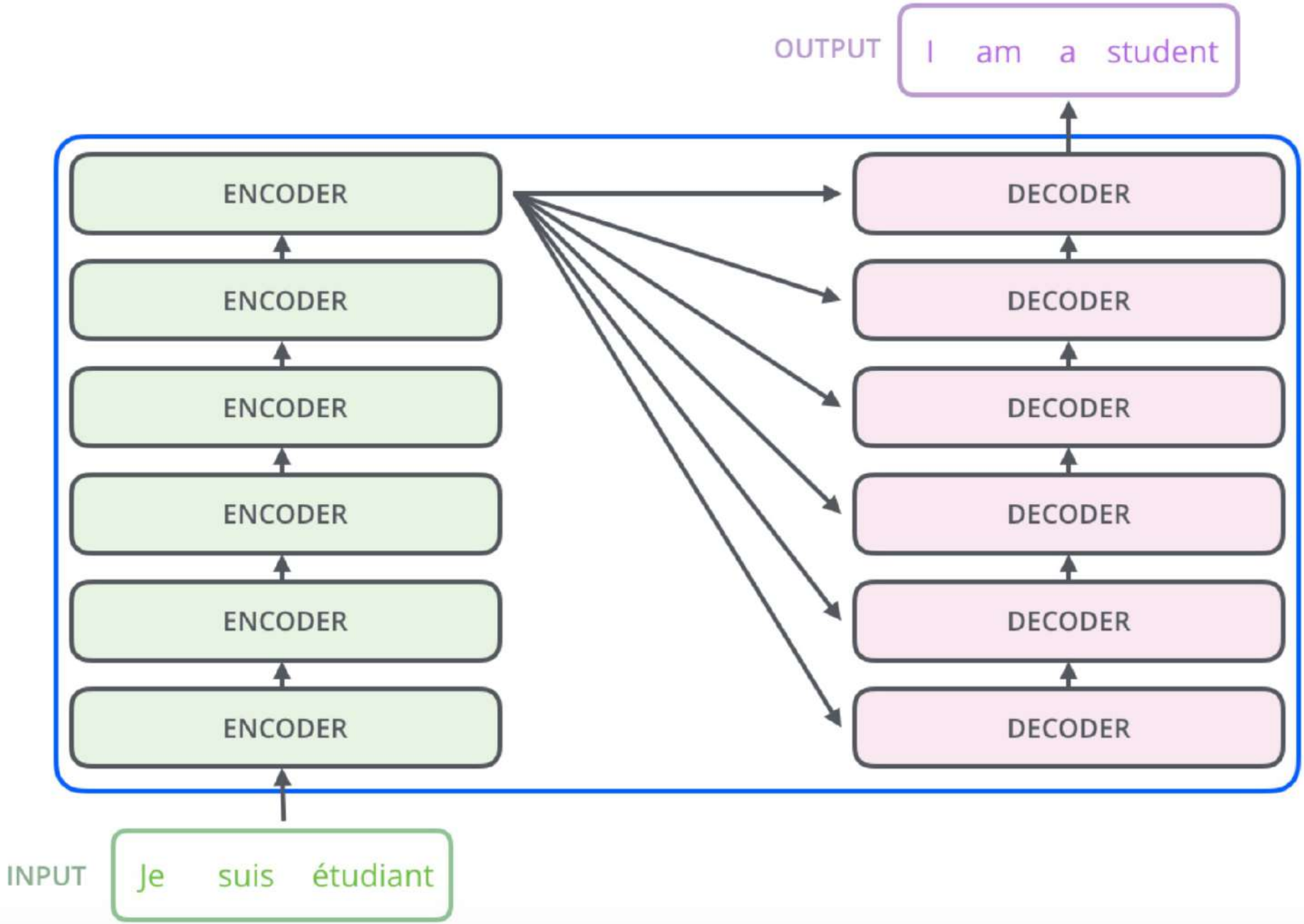



Transformer

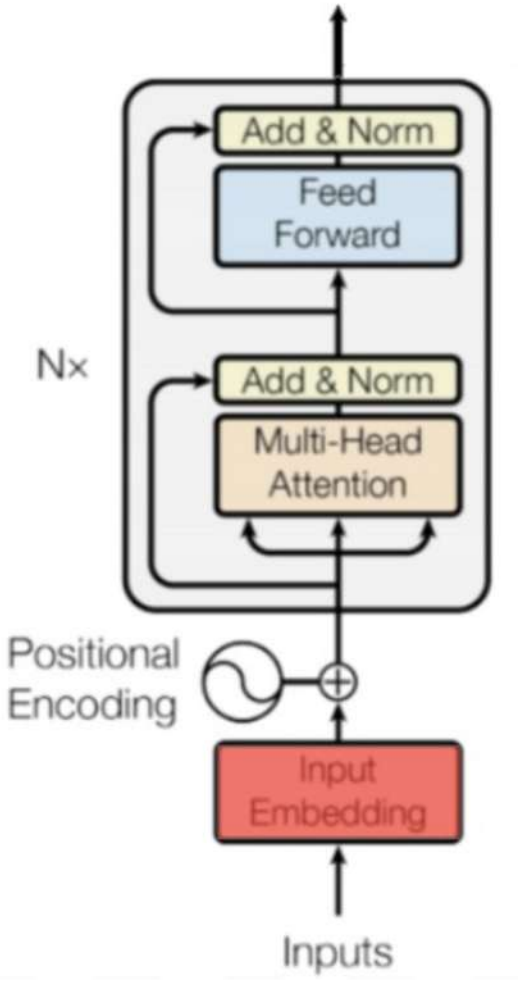


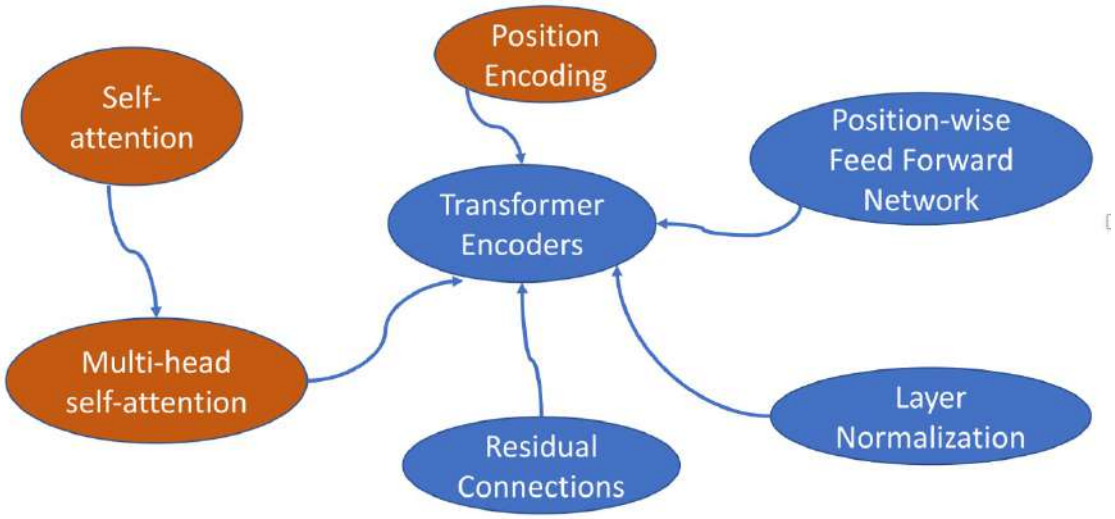


Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.*), Advances in neural information processing systems*. Curran Associates, Inc

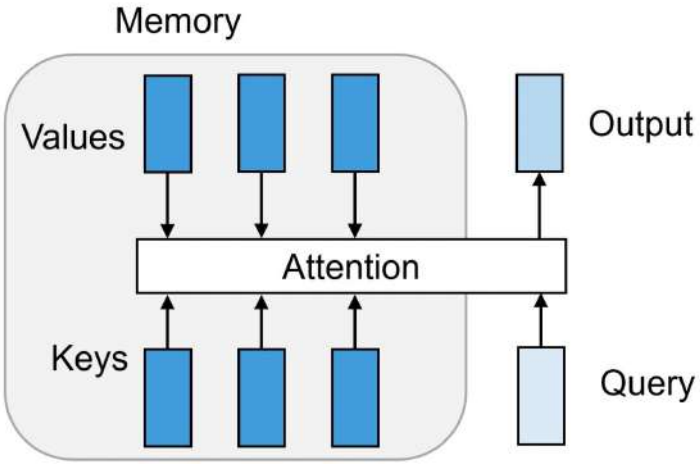

OUTPUT
I am a student
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
DECODER
DECODER
DECODER
DECODER
DECODER
DECODER
INPUT
Je suis étudiant

Inside an Encoder Block

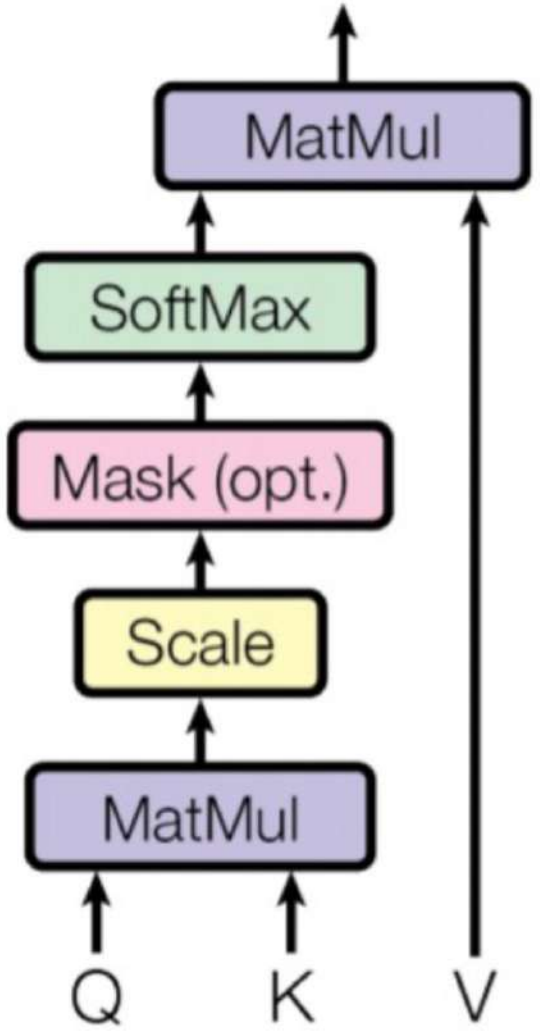




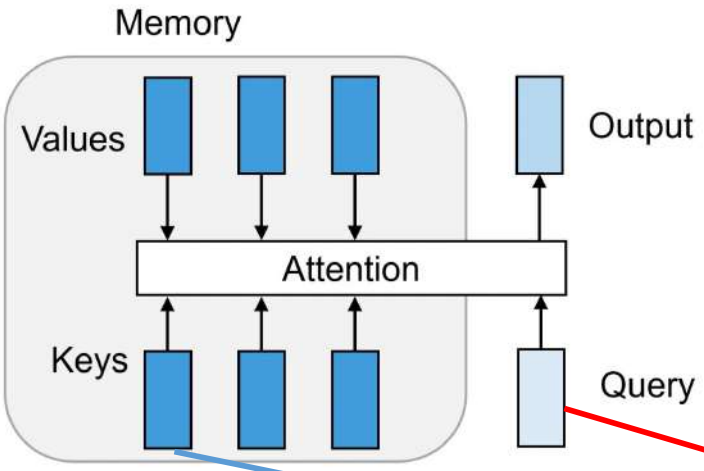
Scaled Dot Product Attention



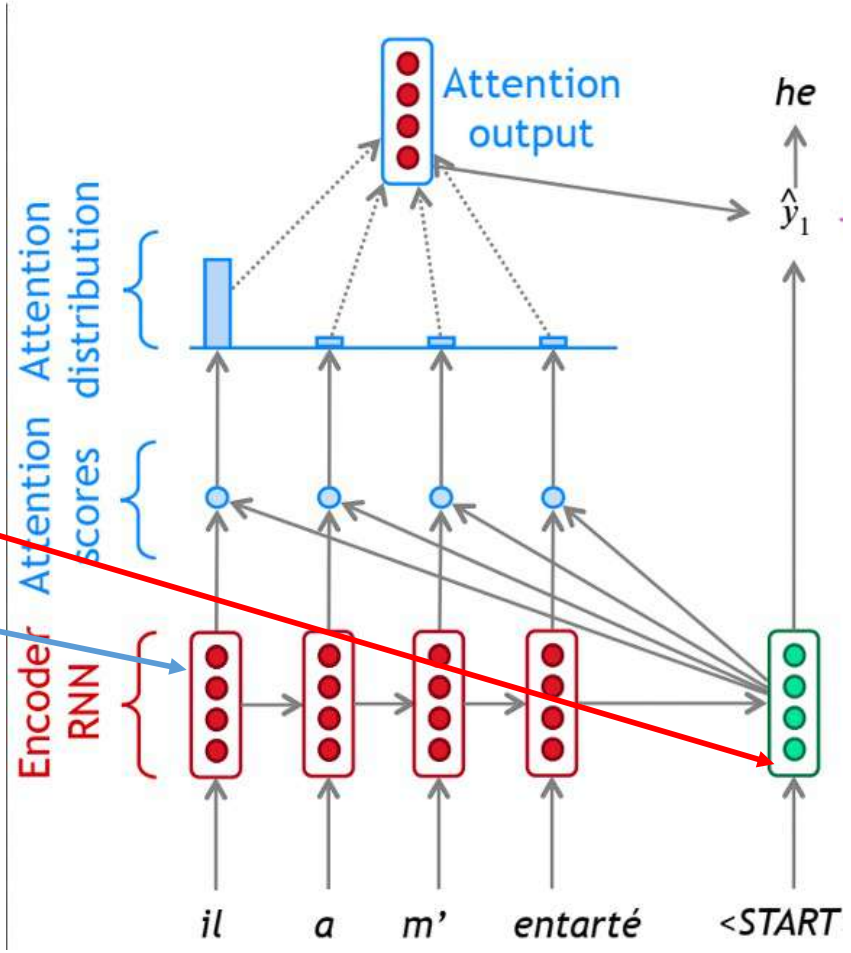
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



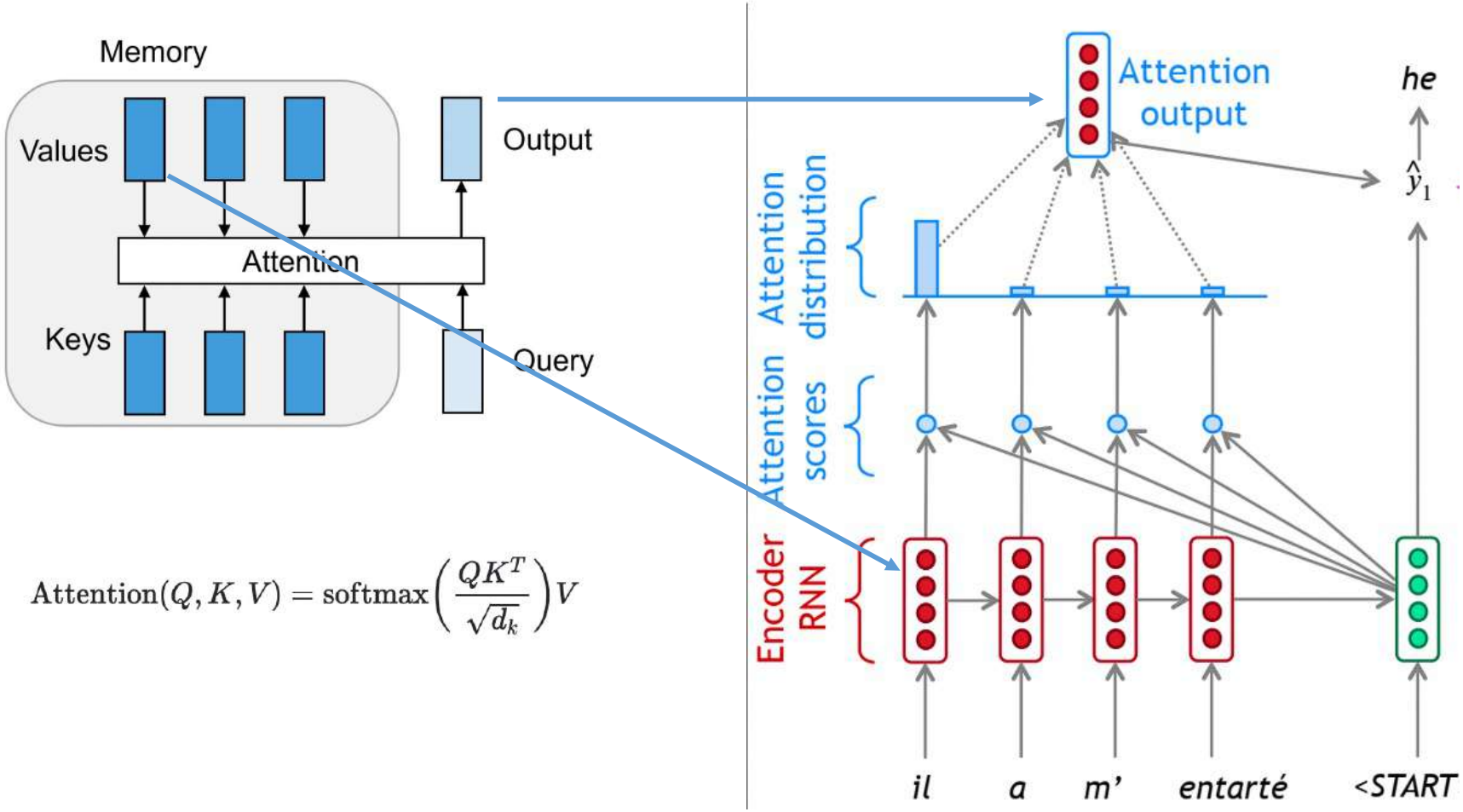
Scaled Dot Product Attention



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

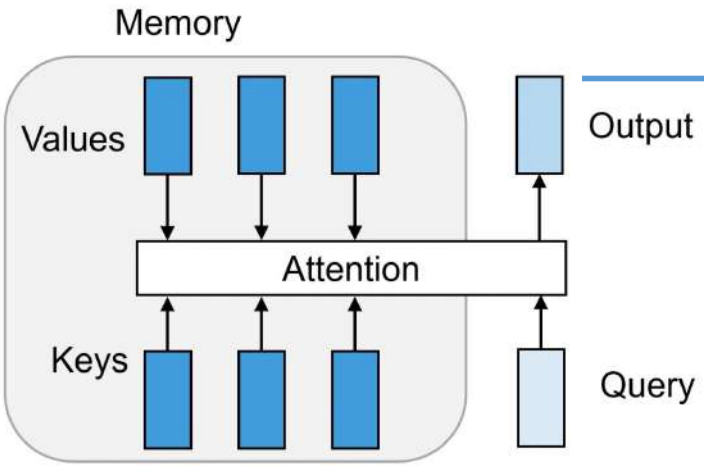


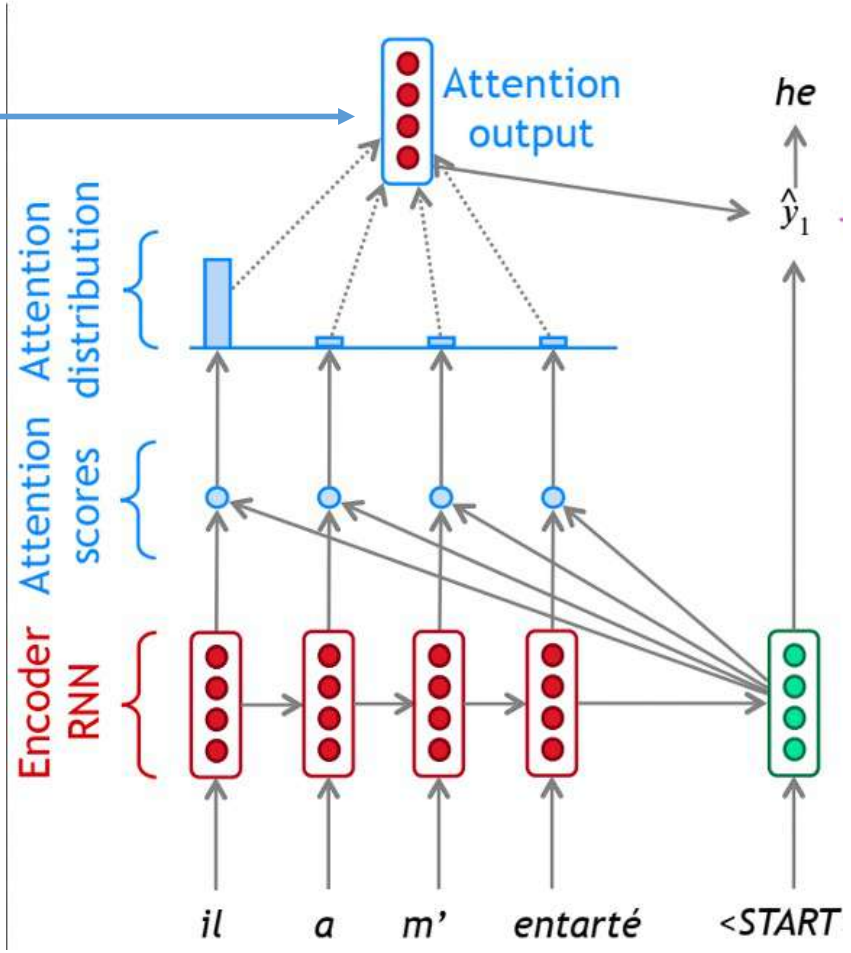
Scaled Dot Product Attention



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

Scaled Dot Product Attention



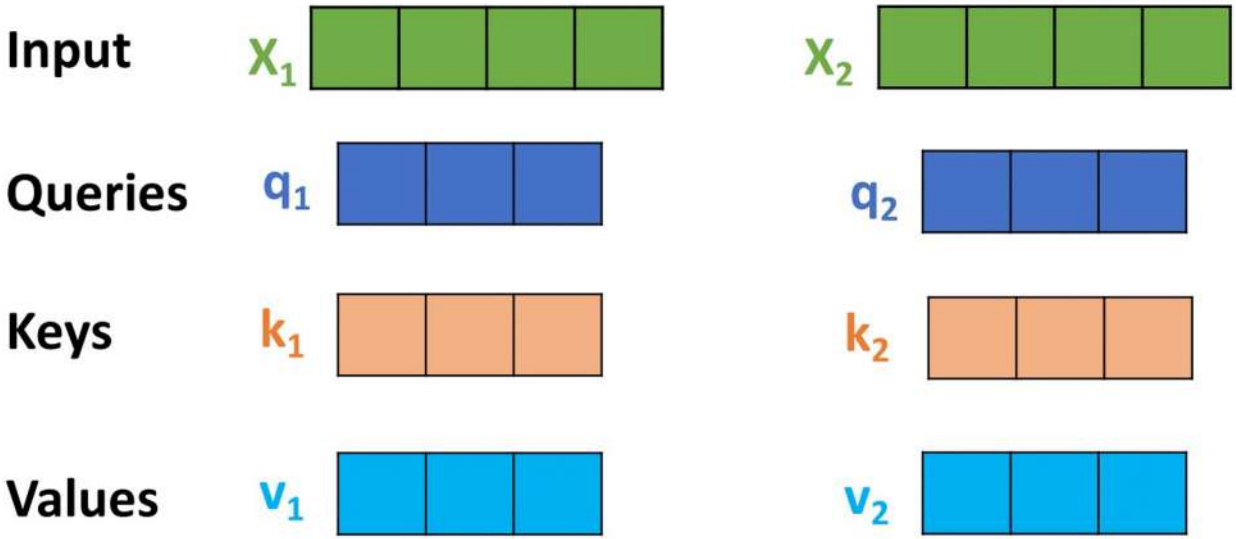


RNN-based Seq2Seq:

- Keys and Values are the same
- Queries are provided from encoder

Self-Attention in Transformer

- Attention maps a query and a set of key-value pairs to an output
  - query, keys, and output are all vectors



Use matrices W^Q, W^K and W^V to project input into query, key and value vectors

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

d_k is the dimension of key vectors

Self-Attention

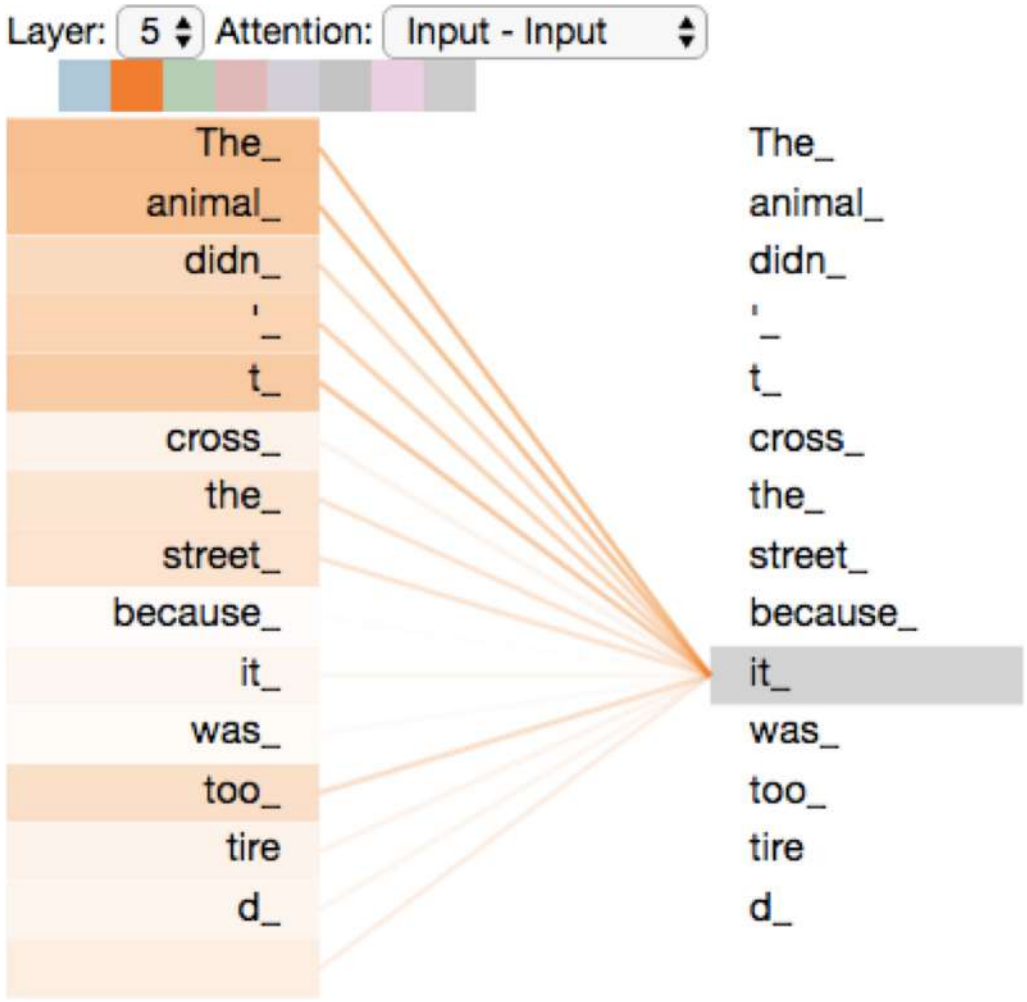
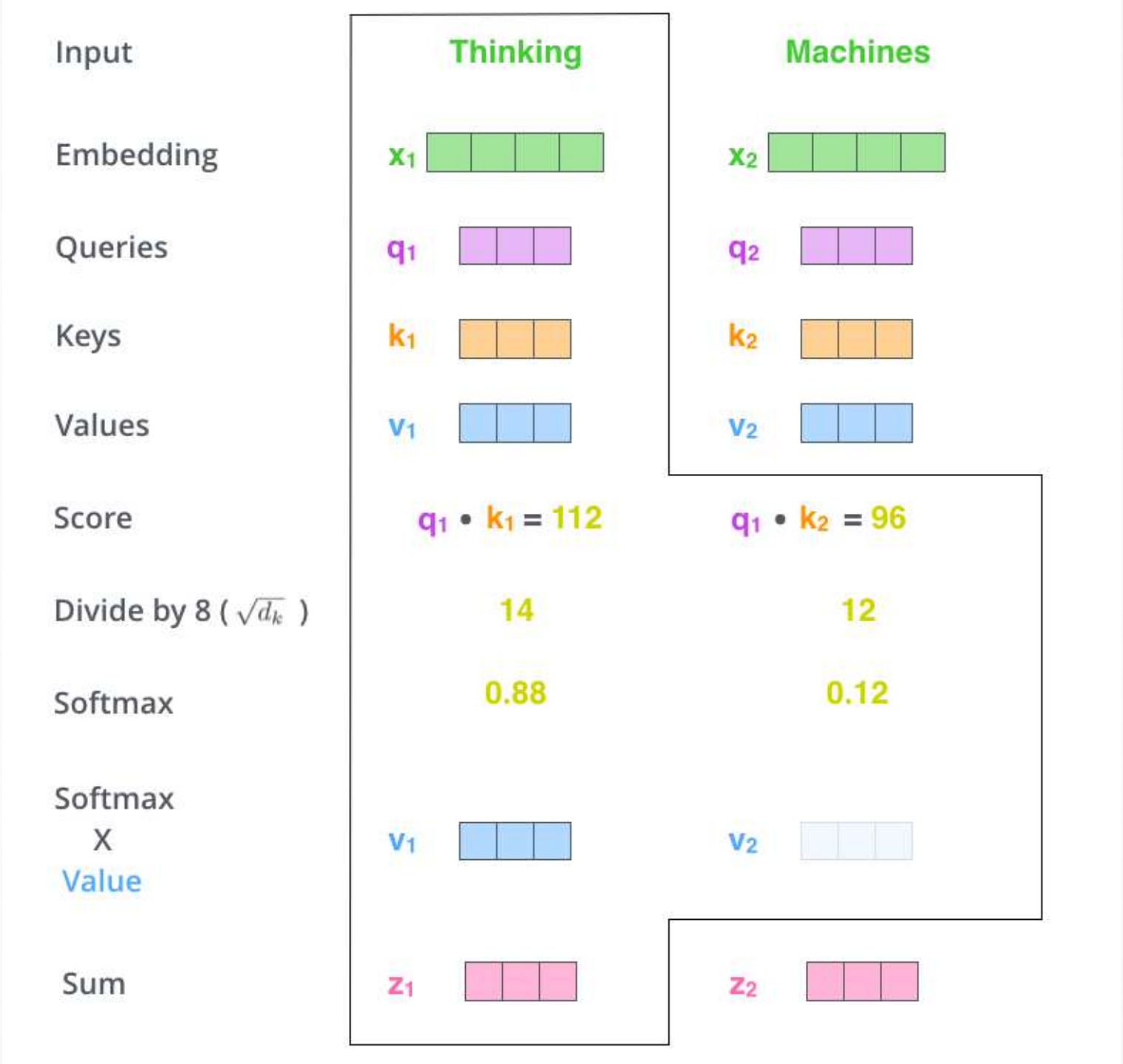
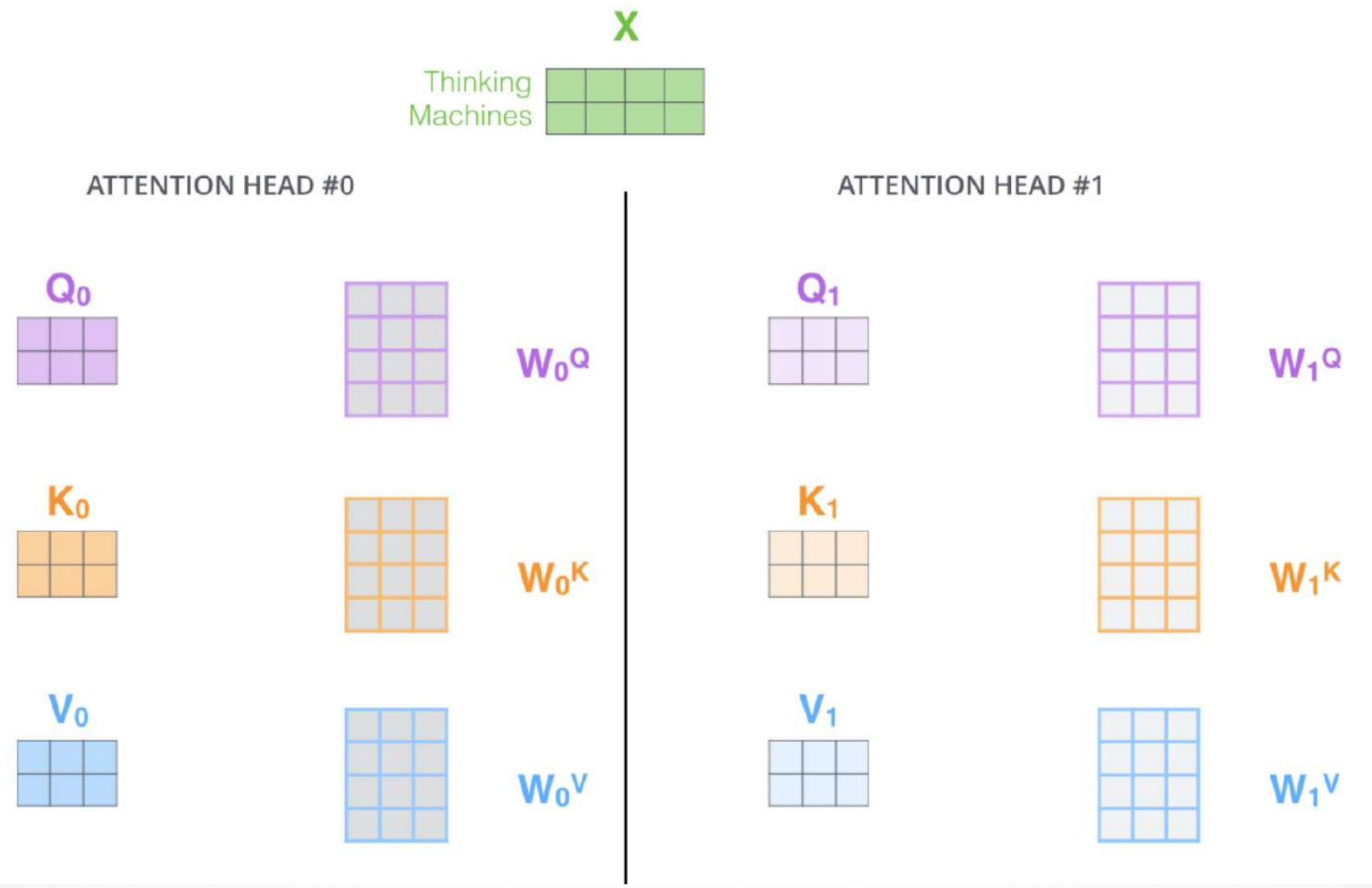
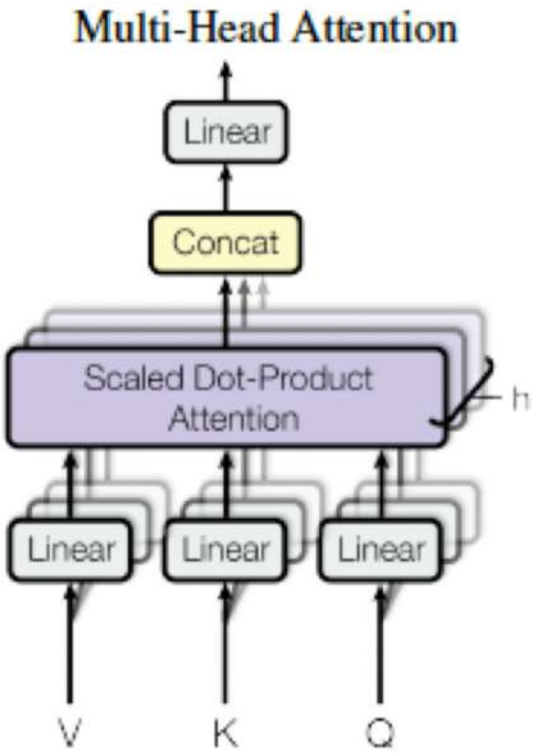


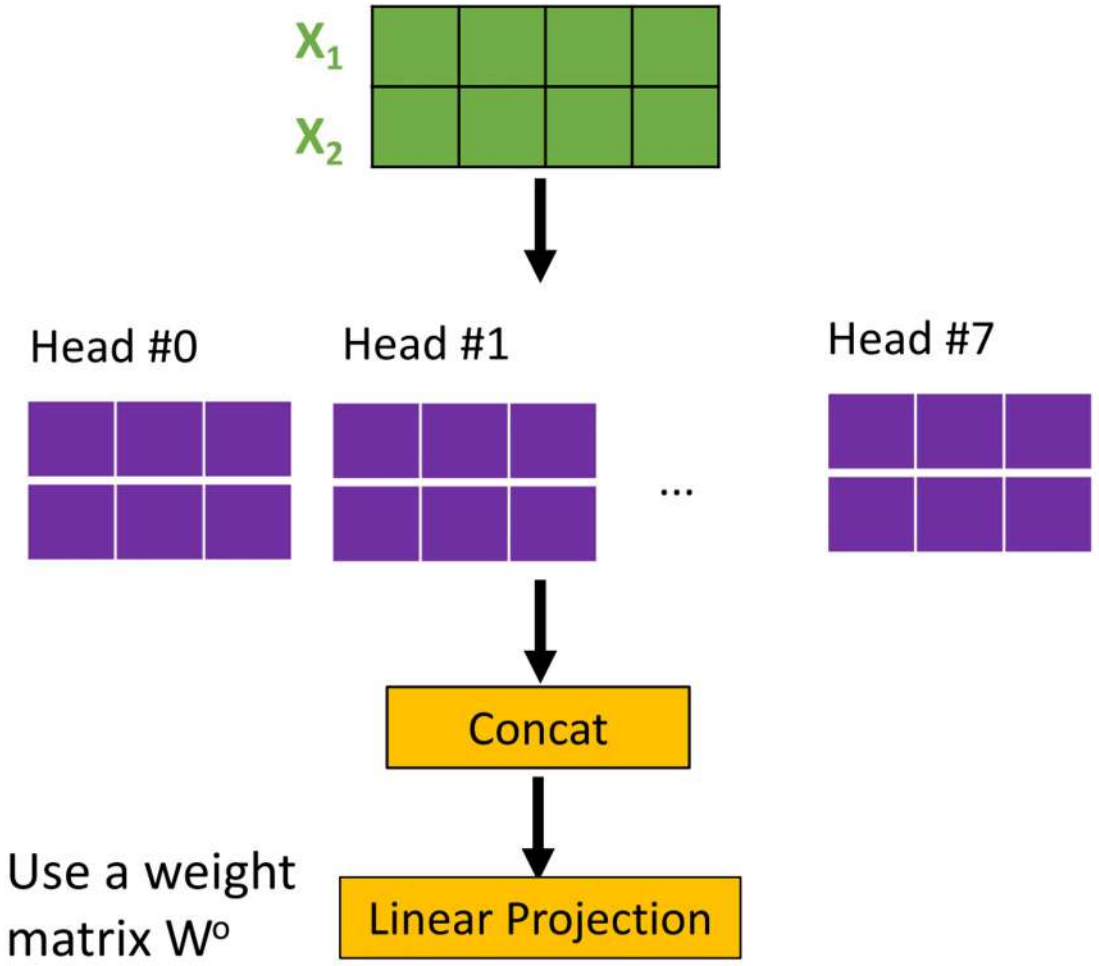
Image source:
https://jalammar.github.io/illustrated-transformer/


Input
Thinking
Machines
Embedding
x_1
x_2
Queries
q_1
q_2
Keys
k_1
k_2
Values
v_1
v_2
Score
$q_1 \cdot k_1 = 112$
$q_1 \cdot k_2 = 96$
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
ATTENTION HEAD #0
ATTENTION HEAD #1
Q0
K0
V0
W0Q
W0K
W0V
Q1
K1
V1
W1Q
W1K
W1V

Multi-Head Attention





1) This is our input sentence*

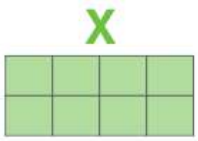
2) We embed each word*

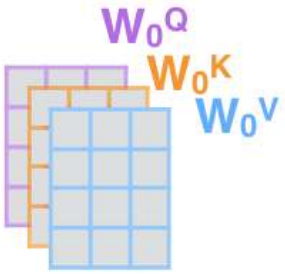
3) Split into 8 heads. We multiply X or R with weight matrices

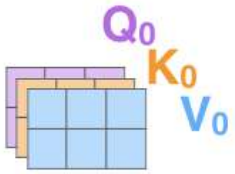
4) Calculate attention using the resulting Q/K/V matrices

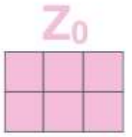
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

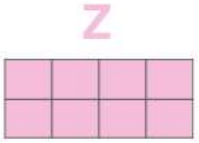
Thinking
Machines



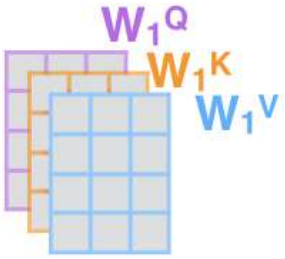


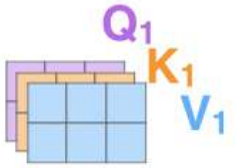


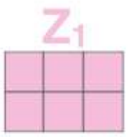


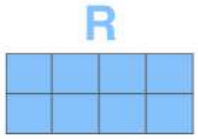


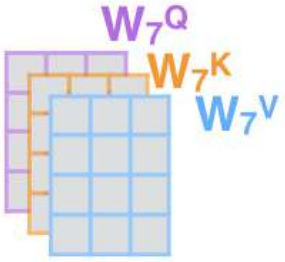
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

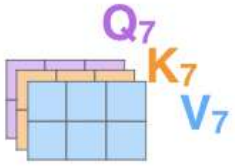


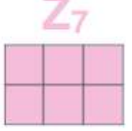












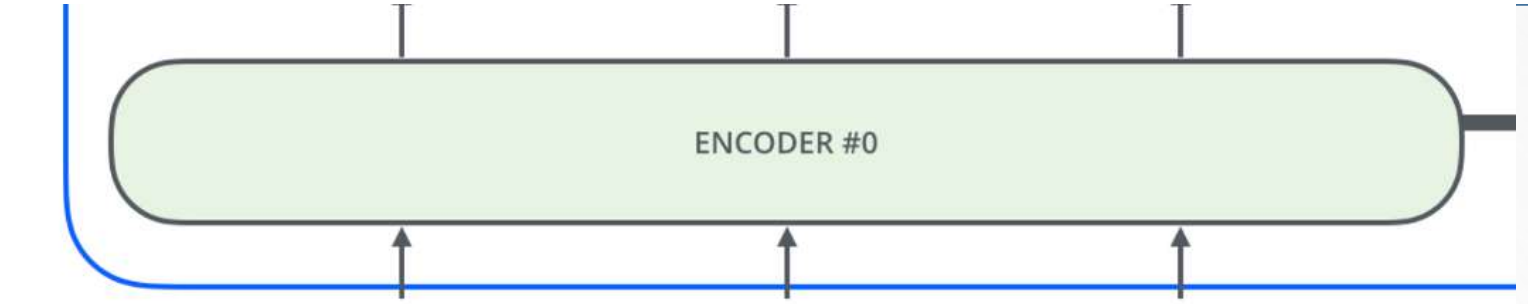
Position Encoding

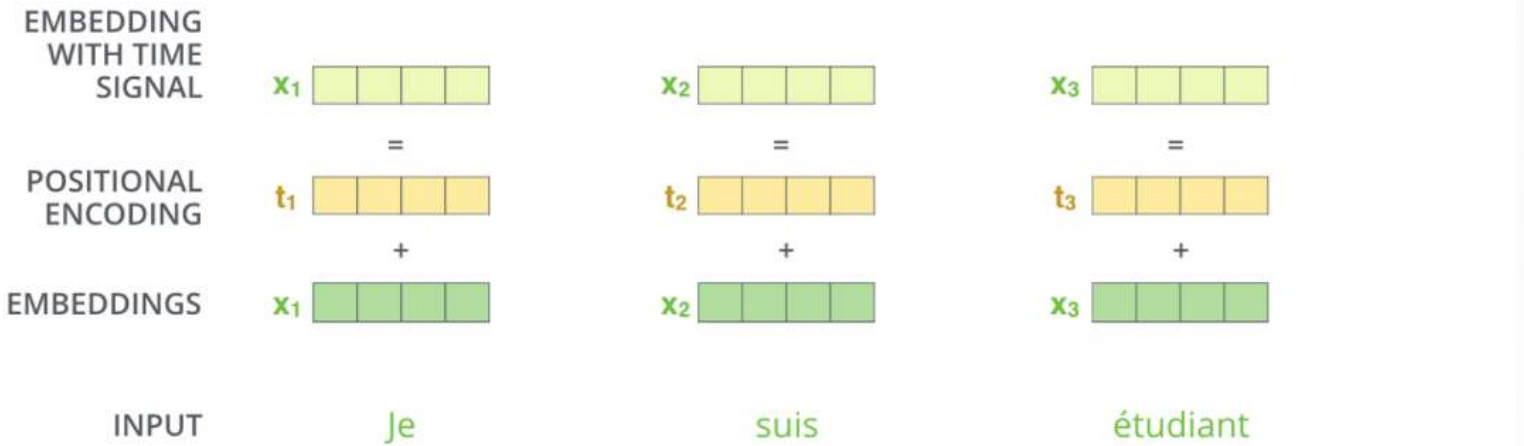
- Position Encoding is used to make use of the order of the sequence
  - Since the model contains no recurrence and no convolution
- In Vawasni et al., 2017, authors used sine and cosine functions of different frequencies

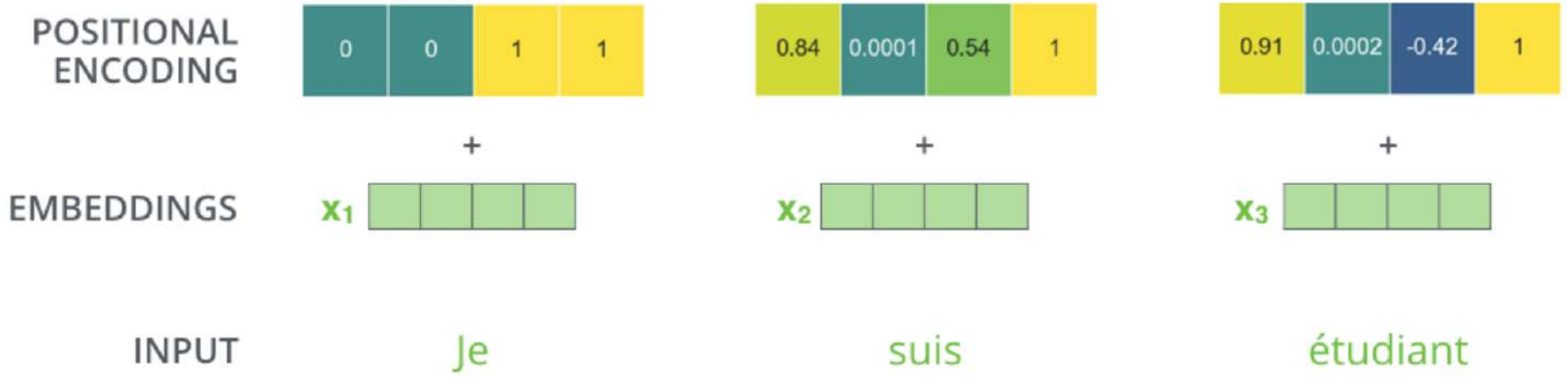
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

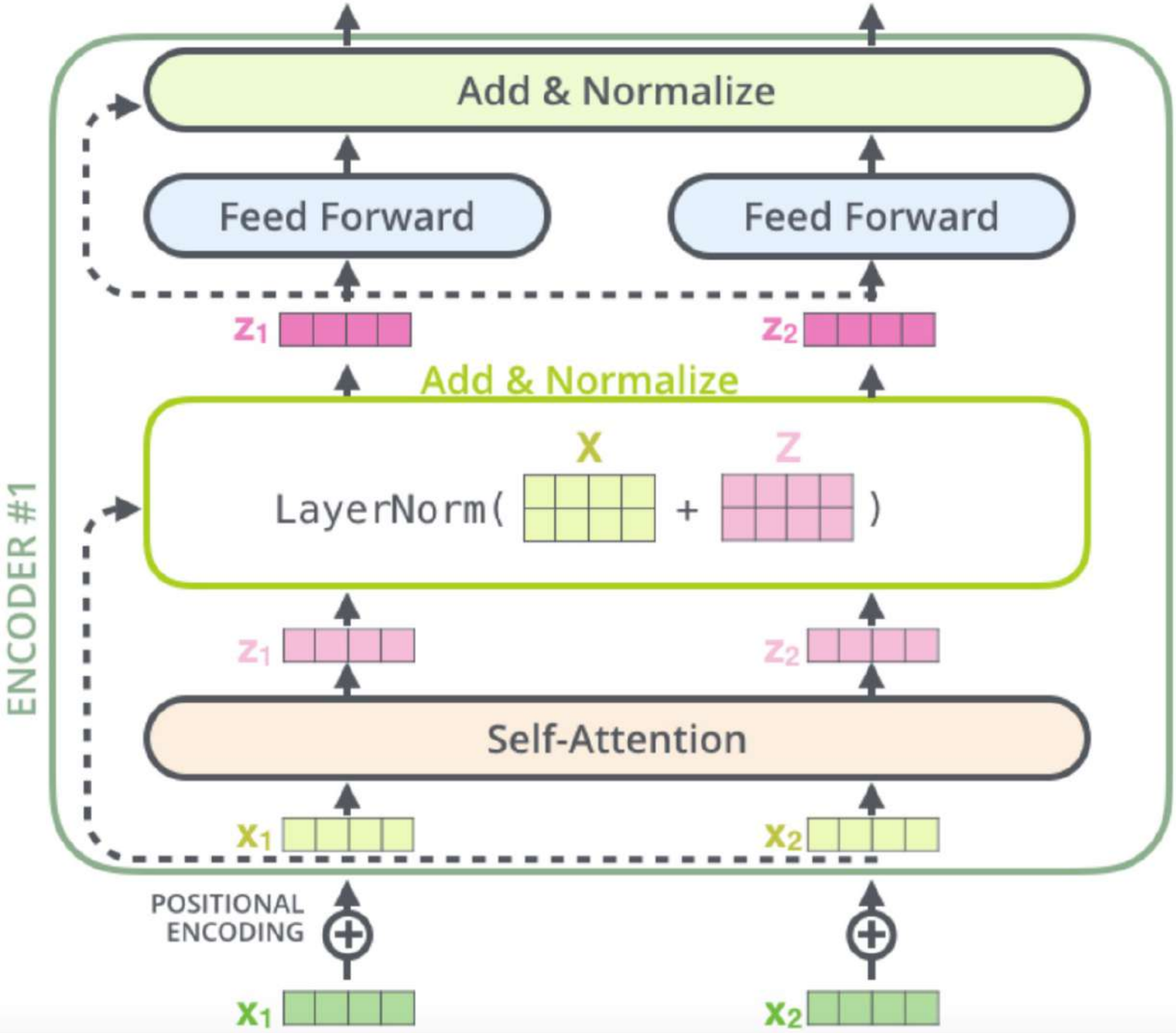
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

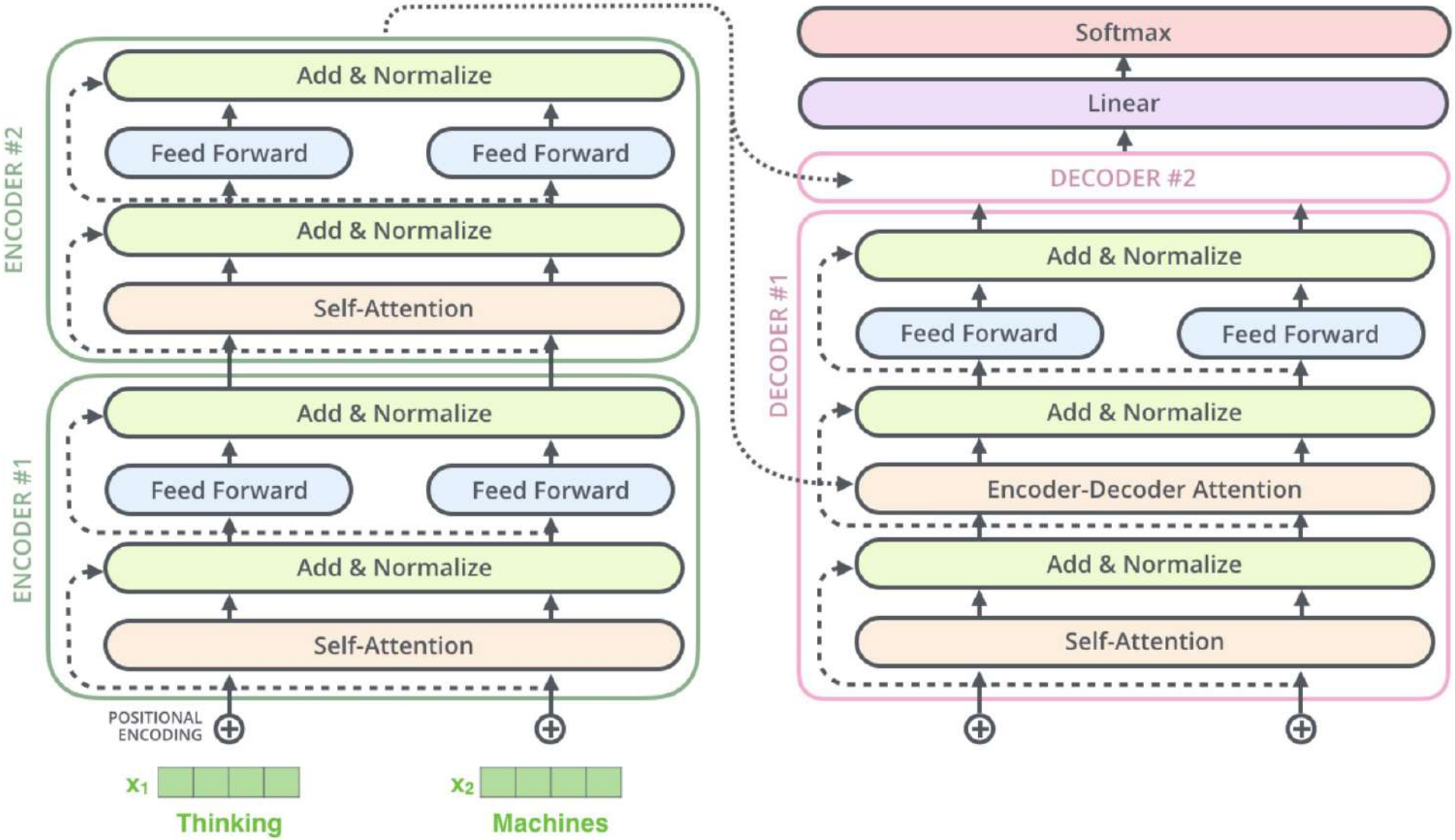
  - *pos* is the position and i is the dimension

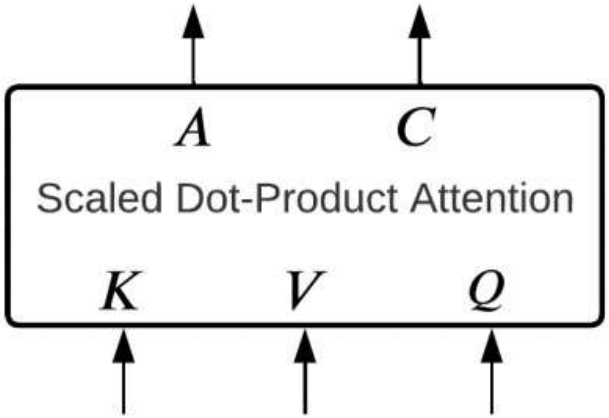

ENCODER #0


EMBEDDING WITH TIME SIGNAL
POSITIONAL ENCODING
EMBEDDINGS
INPUT
x1
x2
x3
t1
t2
t3
=
+
Je
suis
étudiant

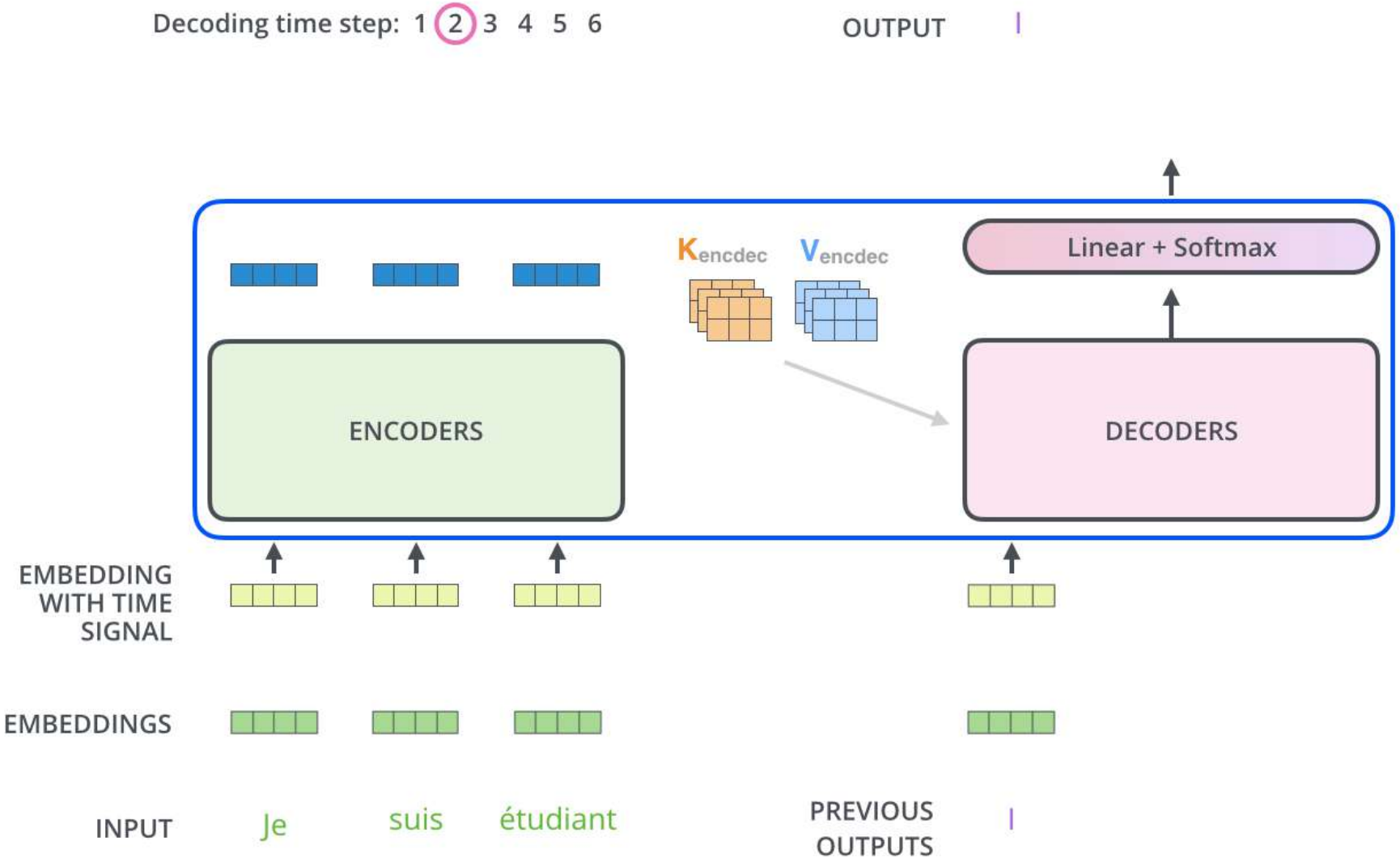

POSITIONAL ENCODING
EMBEDDINGS
INPUT
0
0
1
1
0.84
0.0001
0.54
1
0.91
0.0002
-0.42
1
+
x1
x2
x3
Je
suis
étudiant


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


Softmax
Linear
DECODER #2
DECODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Encoder-Decoder Attention
Add & Normalize
Self-Attention
ENCODER #2
ENCODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
POSITIONAL ENCODING
x1
x2
Thinking
Machines

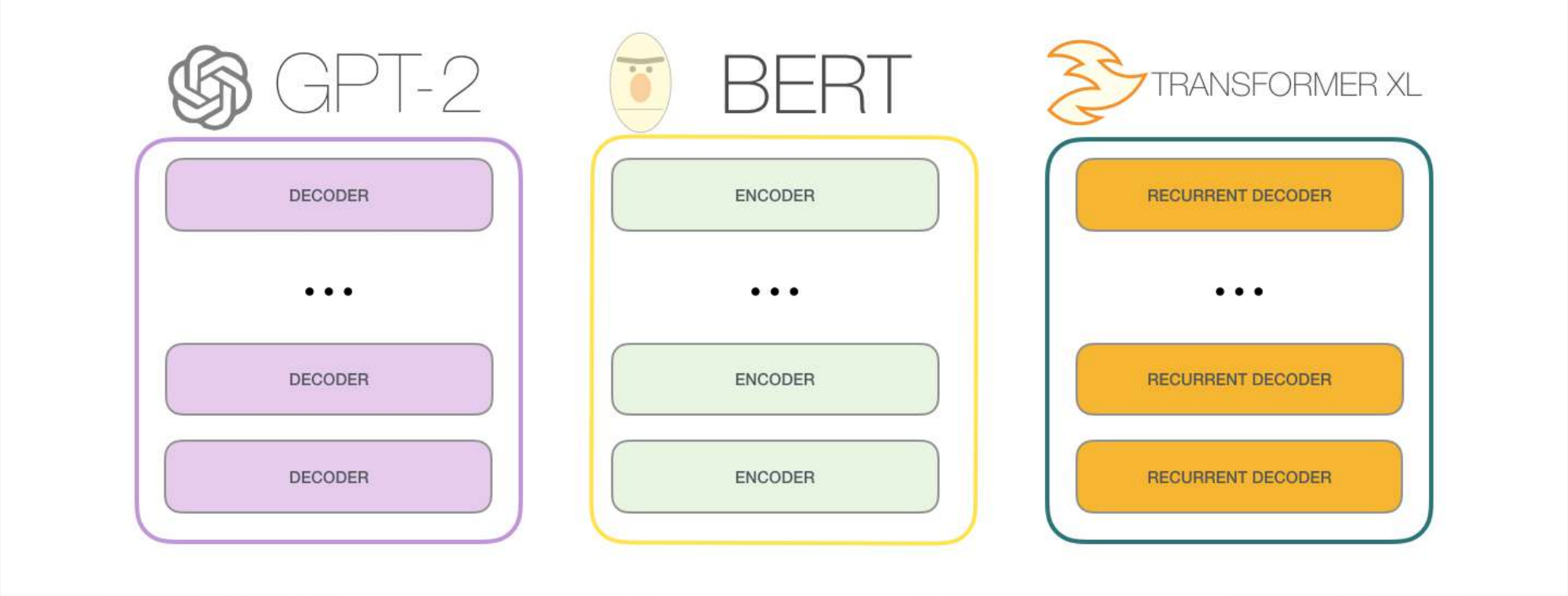

A
C
Scaled Dot-Product Attention
K
V
Q

Vocab_Dist
Softmax
Linear
H_E^D
L x
Add & Norm
FeedForward
Add & Norm
C
Enc-Dec Attention
K V Q
Add & Norm
C
Masked
Self Attention
K V Q
Positional Encoding
Embedding
H_E^L
L x
Add & Norm
FeedForward
Add & Norm
C
self Attention
K V Q
Positional Encoding
Embedding


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

BERT and other variants

Transformer-based Language Models



BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers.
- Use the Transformer Encoder architecture.
- Introduced in 2018 by Google AI.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). *BERT: pretraining of deep bidirectional transformers for language understanding* (J. Burstein, C. Doran, & T. Solorio, Eds.).

unsupervised

1 - ~~Semi-supervised~~ training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

Model:

Dataset:

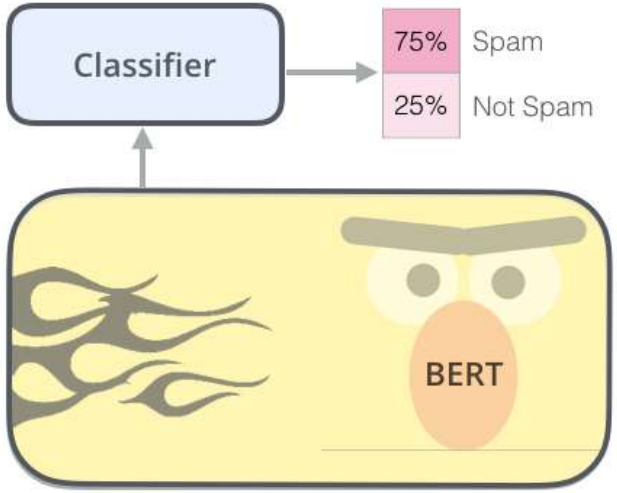
Objective:

Predict the masked word
(langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step

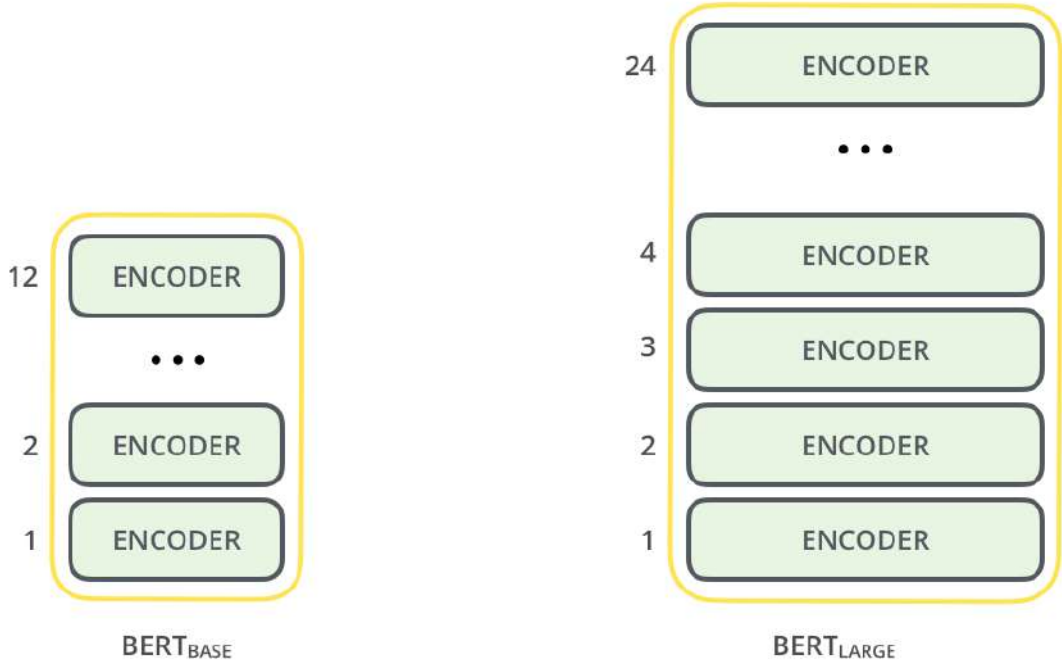
Model:
(pre-trained in step #1)



Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

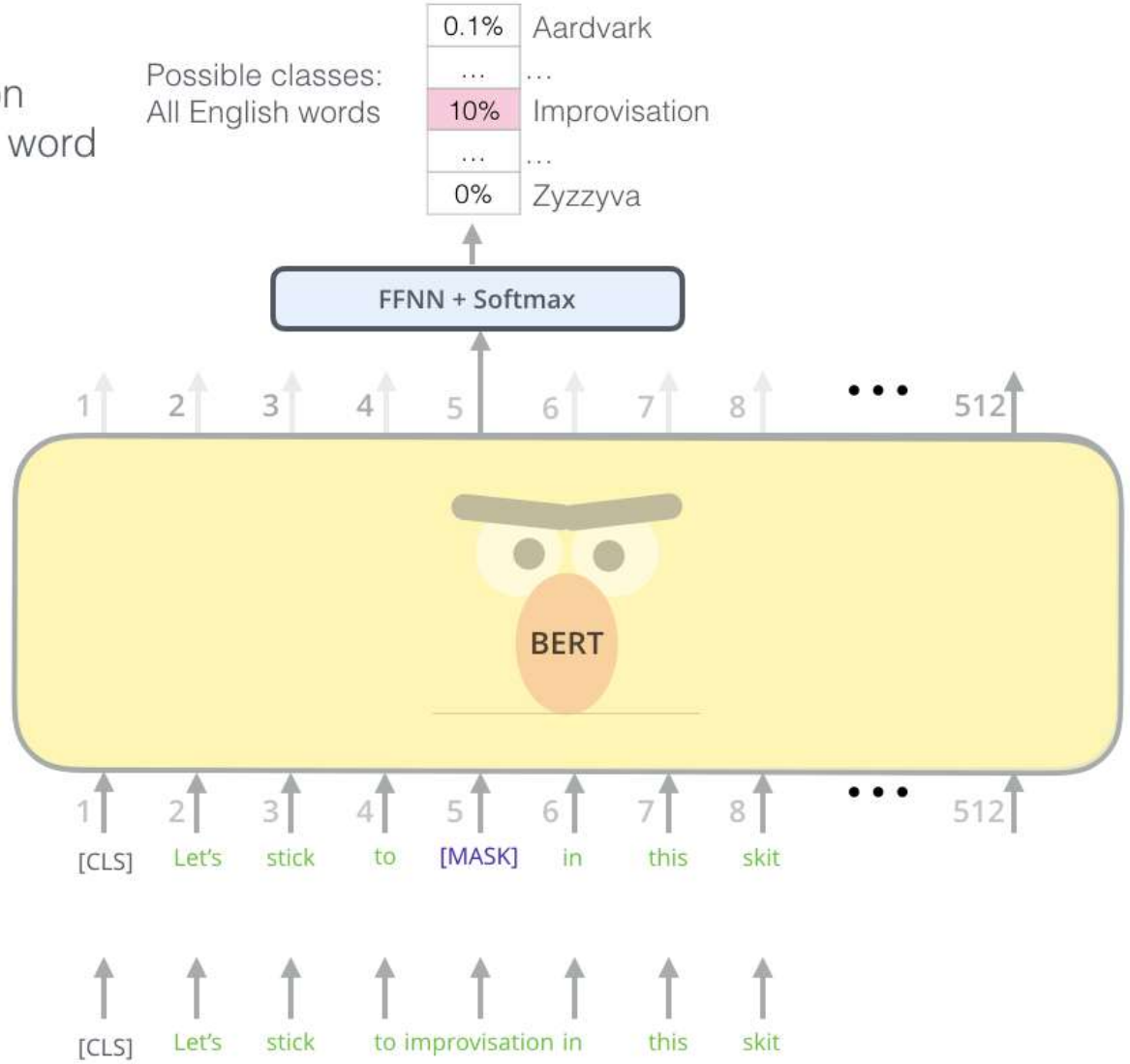
Architecture

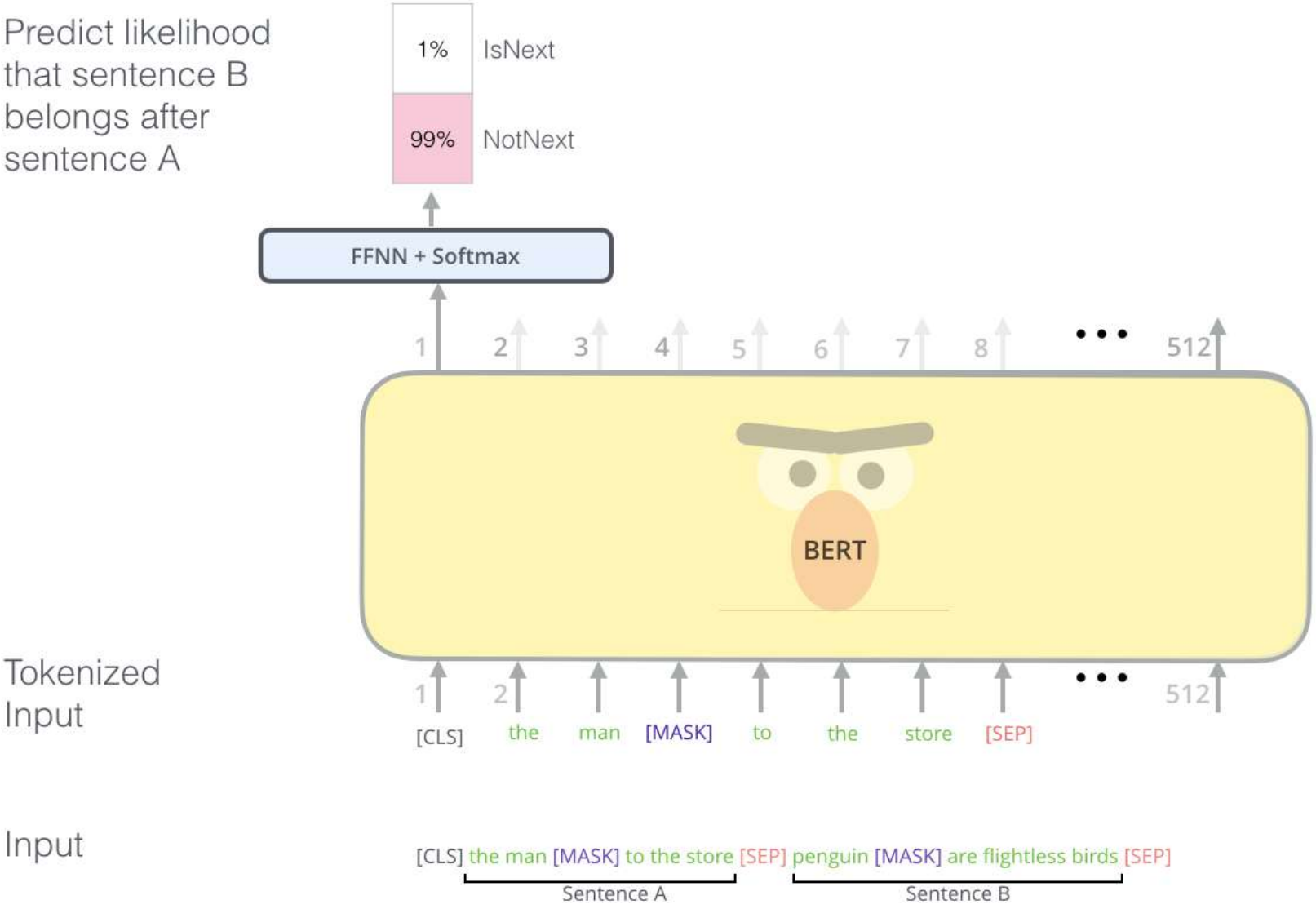


Pretraining

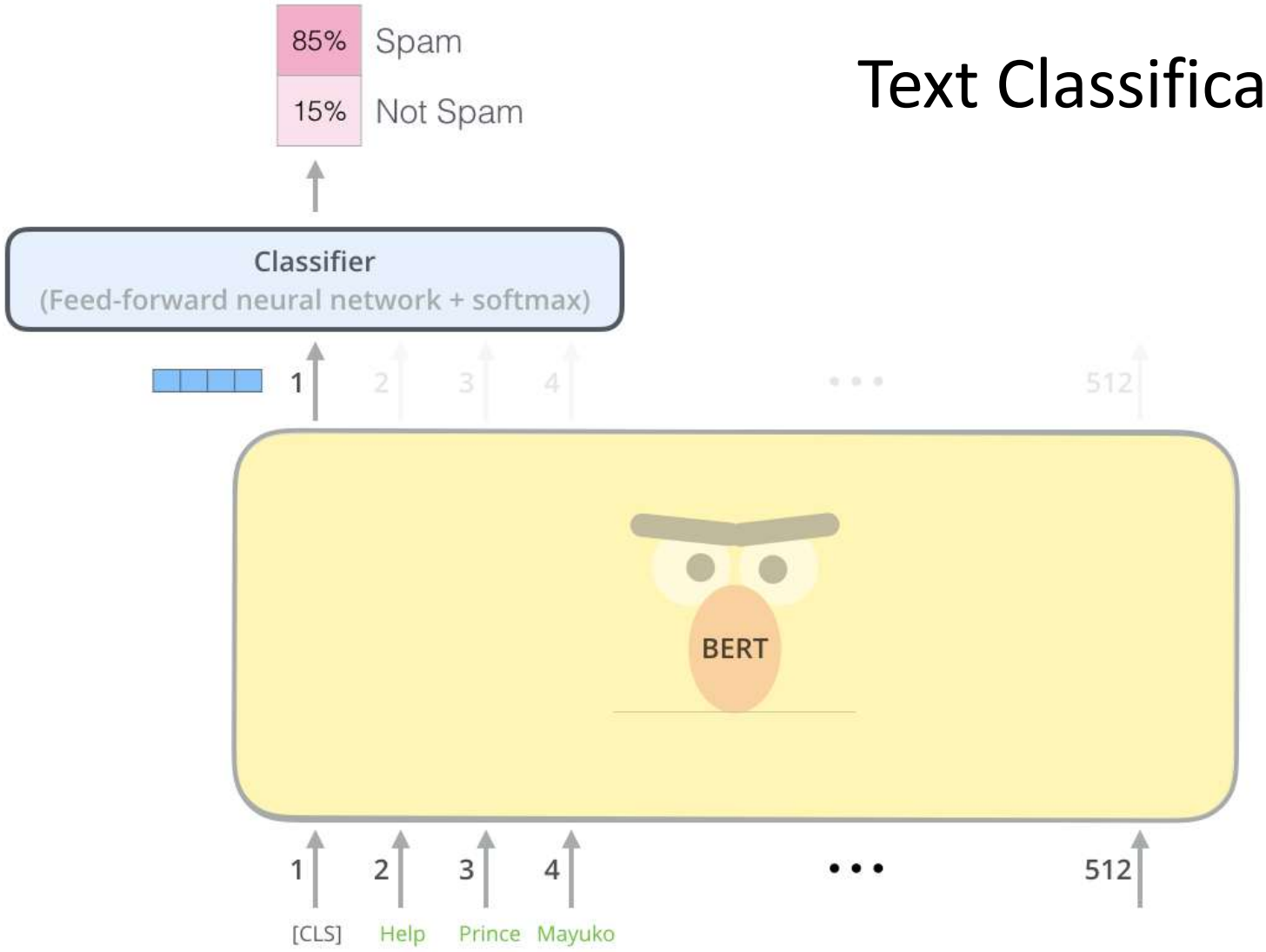
- Two unsupervised tasks:
  1. Masked Language Model
  2. Next Sentence Prediction

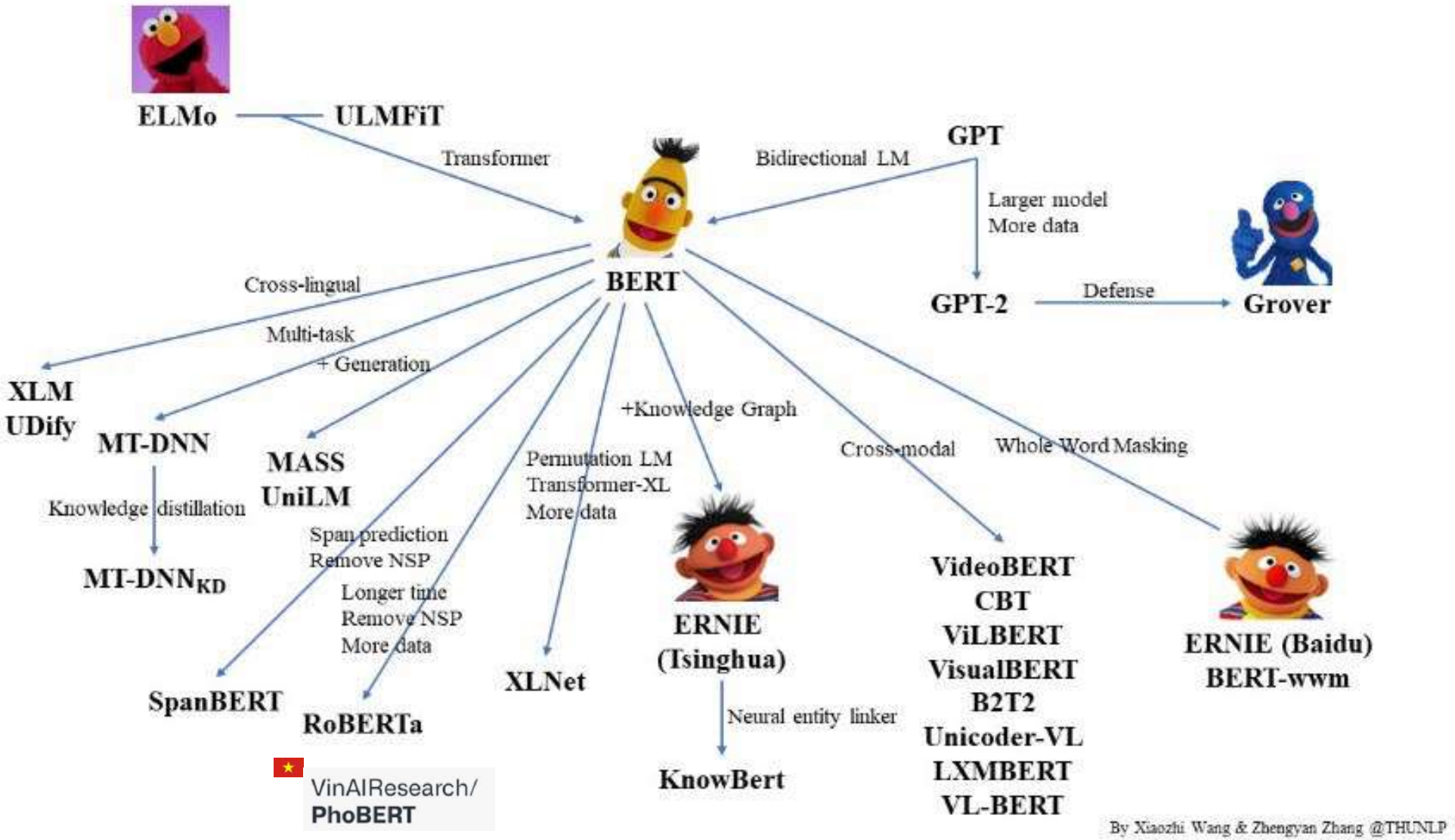
Use the output of the masked word's position to predict the masked word




Predict likelihood that sentence B belongs after sentence A
1%
IsNext
99%
NotNext
FFNN + Softmax
1
2
3
4
5
6
7
8
512
BERT
Tokenized Input
[CLS]
the
man
[MASK]
to
the
store
[SEP]
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Text Classification




ELMo
ULMFiT
Transformer
BERT
Bidirectional LM
GPT
Larger model
More data
GPT-2
Defense
Grover
Cross-lingual
Multi-task
+ Generation
XLM
UDify
MT-DNN
MASS
UniLM
Knowledge distillation
MT-DNN$_{KD}$
Span prediction
Remove NSP
Longer time
Remove NSP
More data
SpanBERT
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMBERT
VL-BERT
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
VinAIResearch/
PhoBERT
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

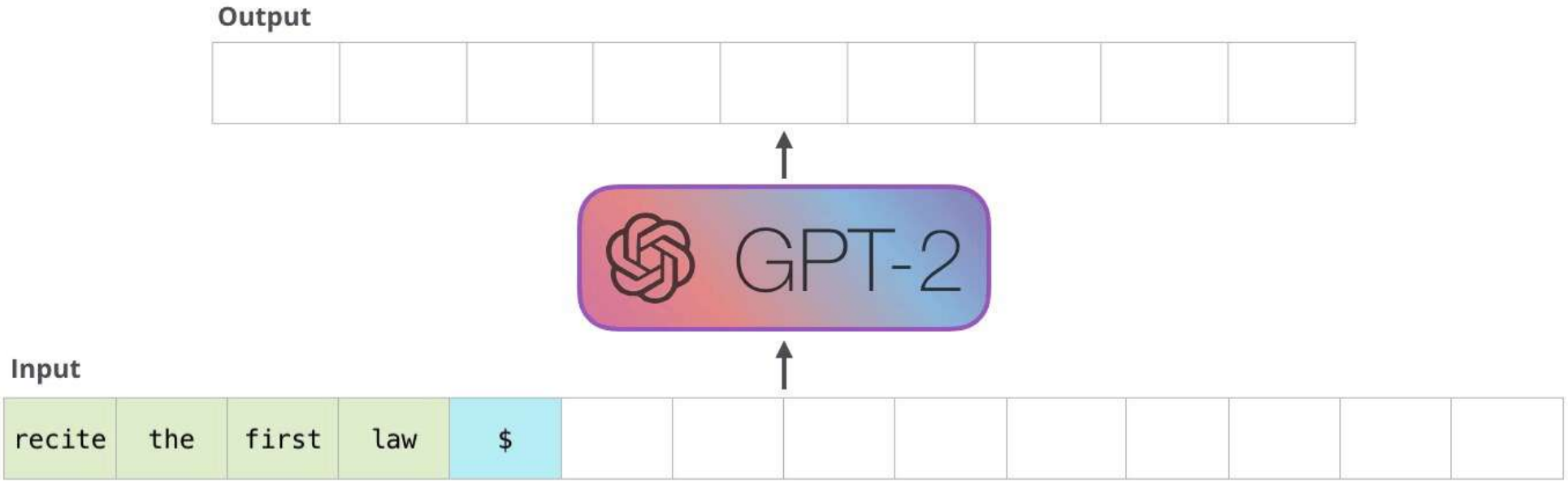
GPT

- Generative Pre-trained Transformer
- Use the Transformer Decoder architecture.
- Introduced in 2018 by OpenAI.

Model	Number of parameters	Training data size	Year
GPT	110M	4GB	2018
GPT-2	1.5B	40GB	2019
GPT-3	175B	≈2TB	2020

Openai [Accessed: 2023-03-01]. (2023). https://openai.com/

How it works?



XLNet

- Autoencoding (BERT):
  - [MASK] tokens do not appear during finetuning ⇒ pretrain-finetuning discrepancy.
  - Assume the predicted tokens are independent of each other given the unmasked tokens. Example: “New York is a city” ⇒ “[MASK] [MASK] is a city”
- Autoregressive (GPT):
  - Only trained to encode a unidirectional context (forward or backward).

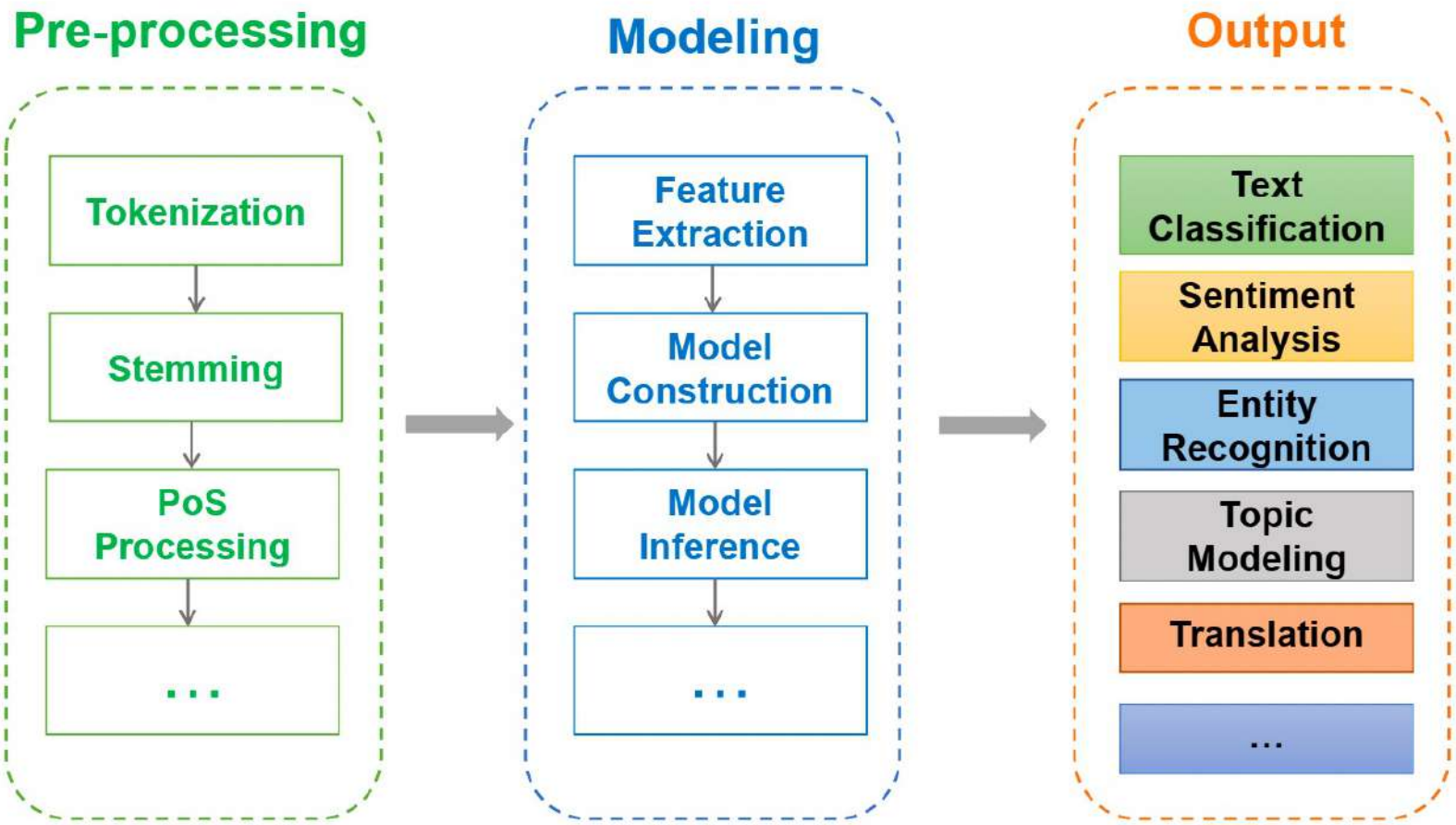
Yang, Z. et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” *NeurIPS* (2019)

XLNet

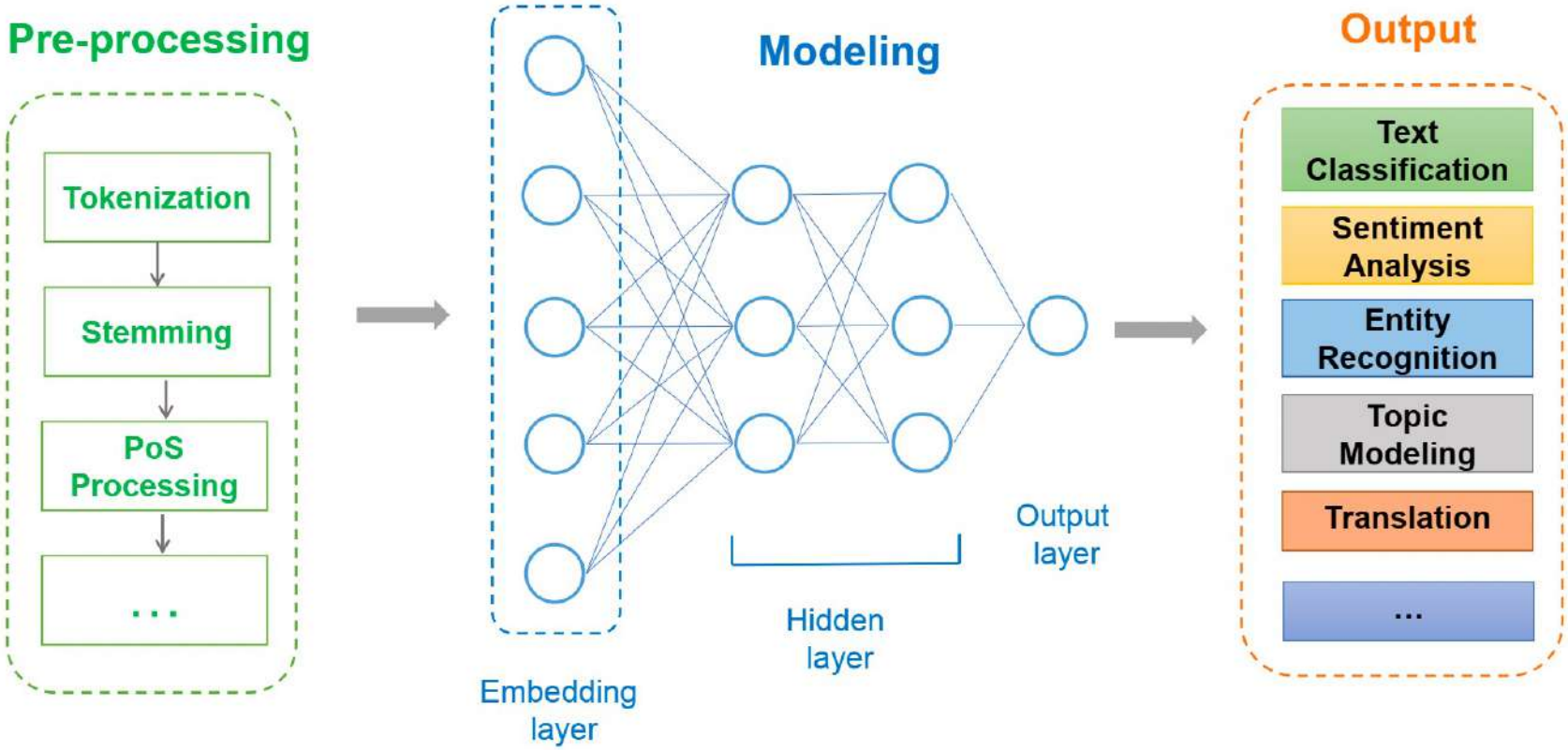
- XLNet combines pros from both while avoiding their cons.
- Techniques:
  - Permutation Language Modeling
  - Two-Stream Self-Attention for Target-Aware Representations
  - Incorporating Ideas from Transformer-XL
  - Modeling Multiple Segments

Applications in NLP

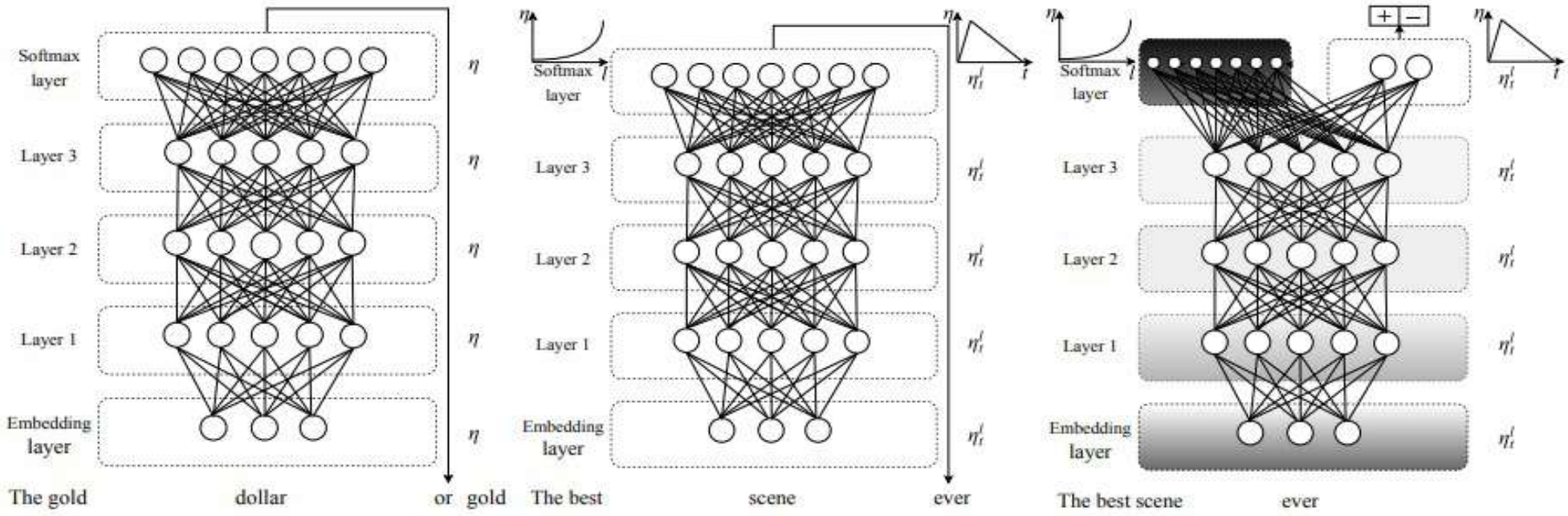
NLP typical pipeline

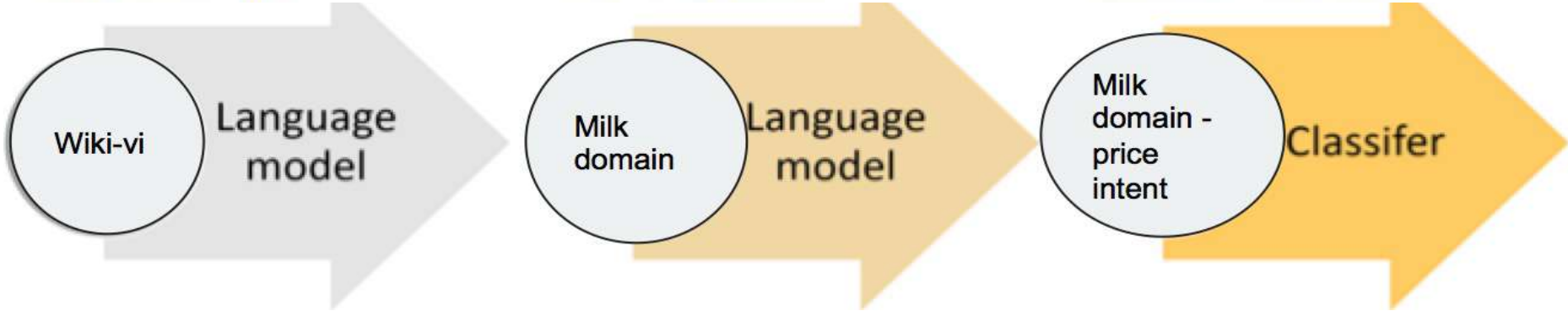


NLP DL-based pipeline



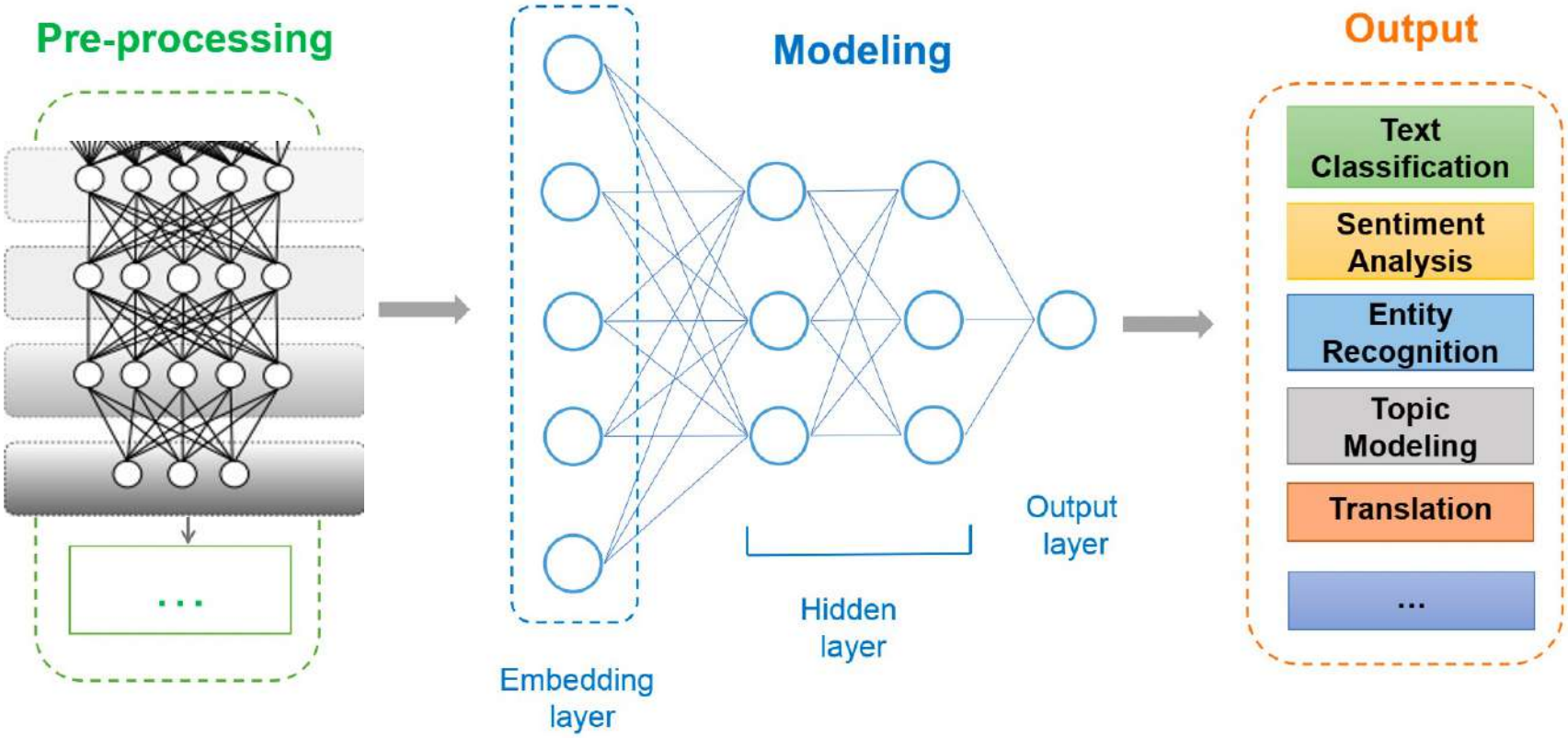
Pre-trained Neural Language Model



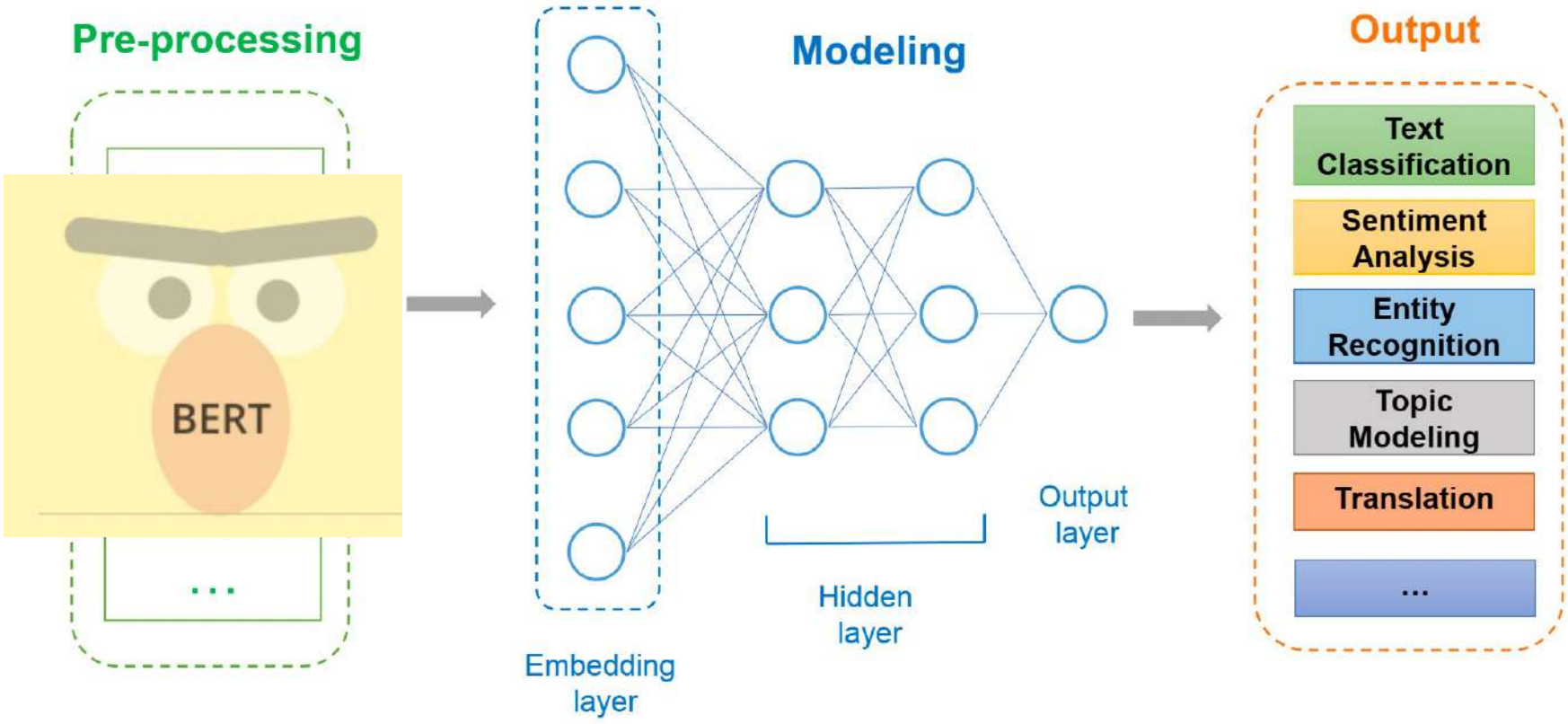


ULMFit (Howard and Rudder, 2018)

NLP LM-based pipeline

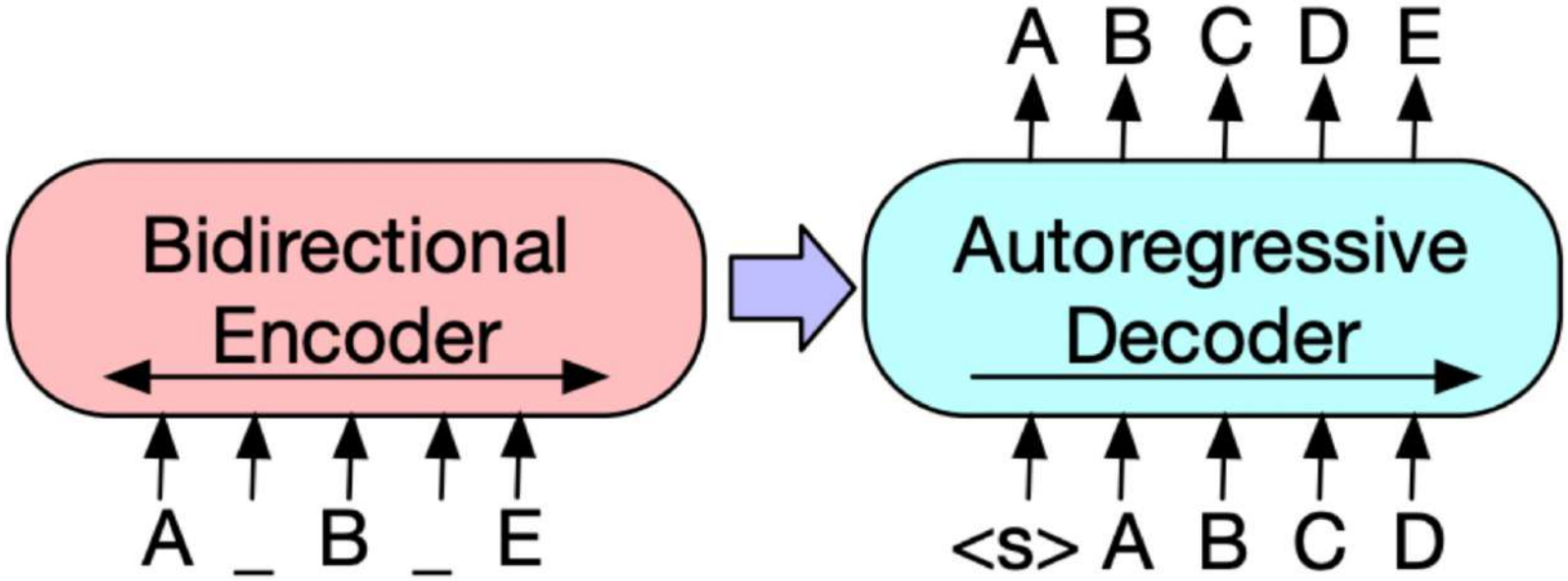


NLP LM-based pipeline



From BERT to BART

- BERT is not a fully Seq2Seq model (i.e. not a generative model)
- BART is introduced as an extended/complement



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

From PhoBERT to BARTPho

VinAIResearch/
BARTpho

BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese (INTERSPEECH 2022)

1 Contributor | 0 Issues | 75 Stars | 6 Forks

BARTPho for Vietnamese translation applications

- Pretrained with Vietnamese
- Implicitly processing “aligning” task
- More powerful if the target language has similar language to Vietnamese (Chinese, Bahnaric, etc.)

A Demo to be concluded

- https://www.ura.hcmut.edu.vn/bahnar/nmt

Thank you