Seq2Seq & Transformer Model

Indonesia AI

OBJECTIVE & OUTLINE

Seq2Seq & Transformer Model

Objektif: Memahami konsep dari Seq2Seq & Transformer Model dalam NLP

Outline:

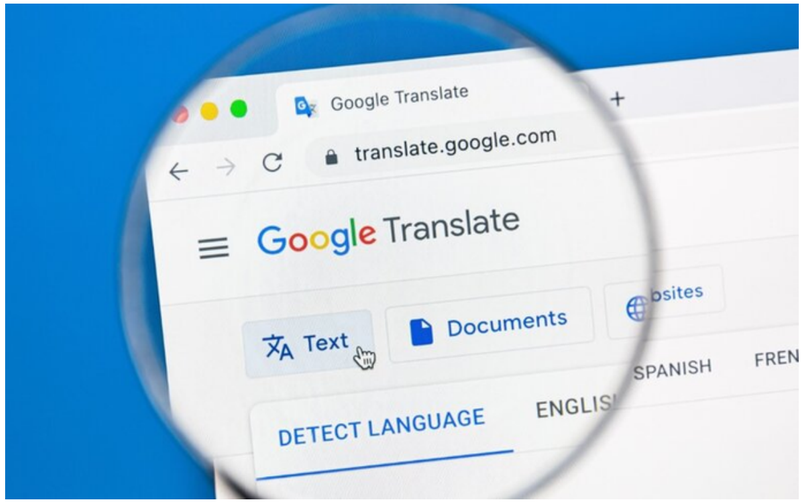
1. Seq2Seq Modl
2. Attention Model
3. Inside Transformer
4. Transformers Development

Seq2Seq Model

Indonesia AI

EXAMPLE

das Haus ist gross →



→ the house is big

EXAMPLE

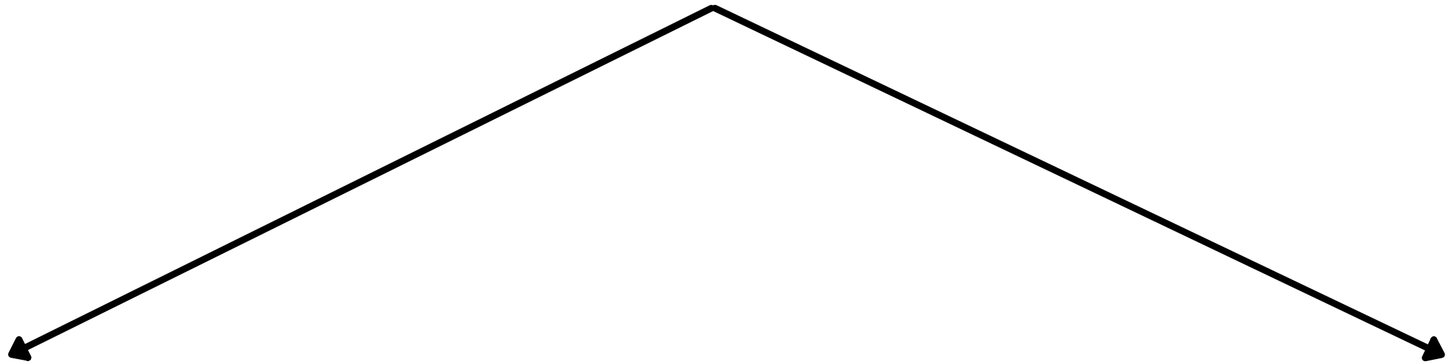
Albert lives in Barlimore →  → PER NONE NONE LOC

NER

HANDLING SEQUENCES

SEQ2SEQ

Architecture



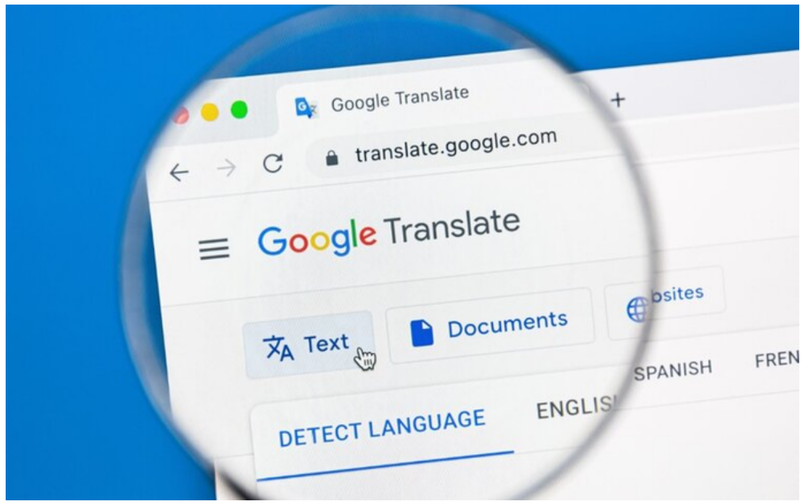
INPUT

Encoder: Convert arbitrary length input to some fixed-length hidden representation

OUTPUT

Decoder: generate arbitrary length output

EXAMPLE

das Haus ist gross → 

→ the house is big

Encoder

das Haus ist gross →

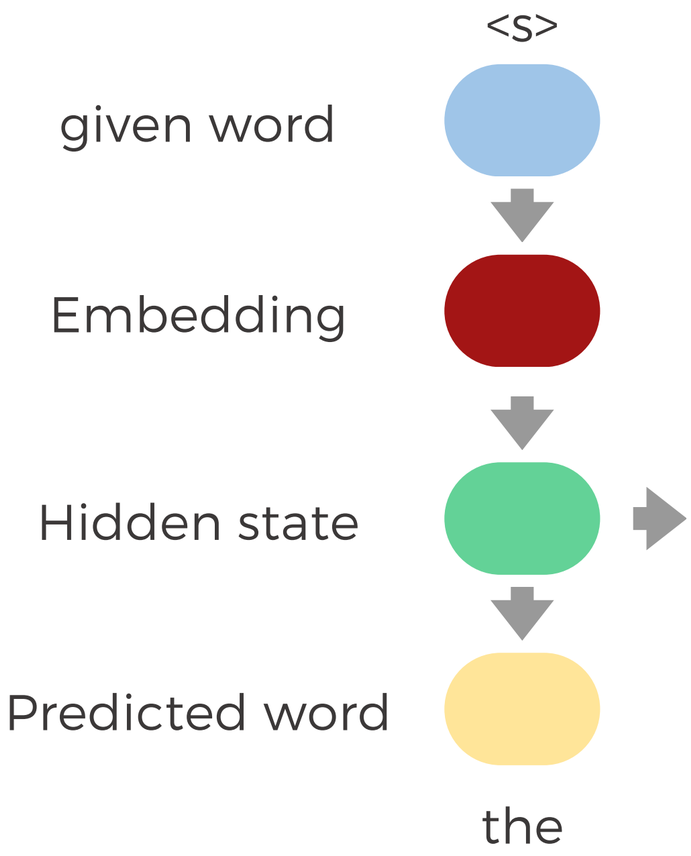
Sentence Vector

Decoder →

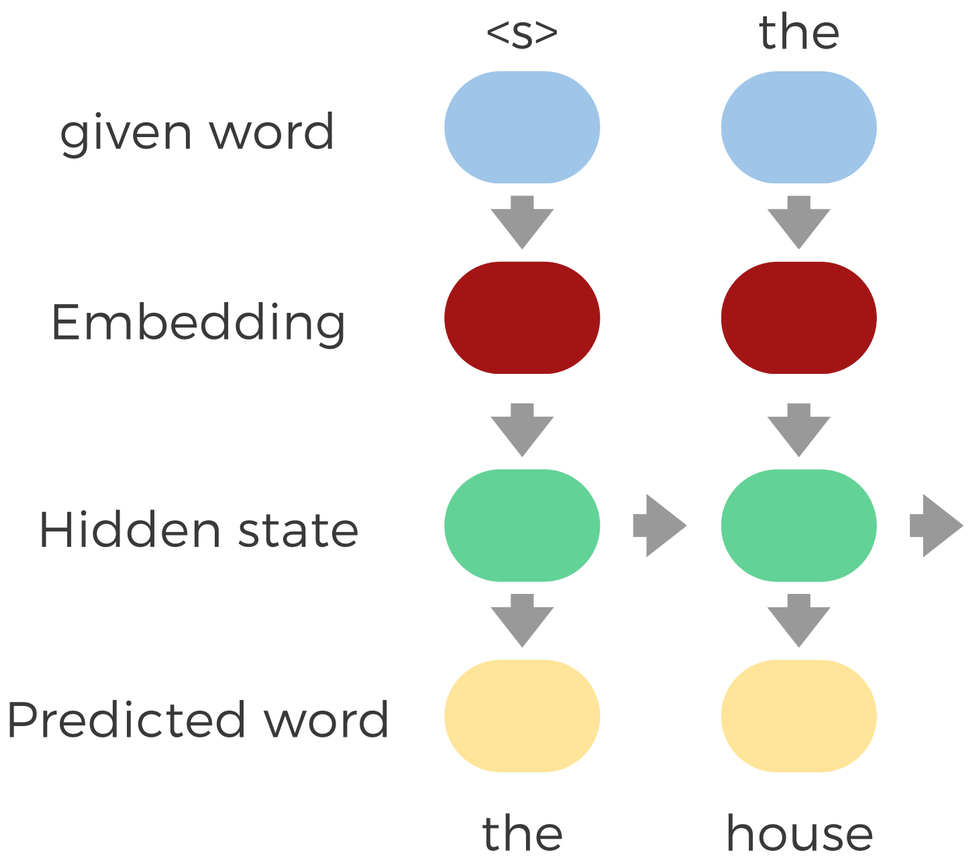
step 1: the
step 2: house
step 3: is
step 4: big
step 5: <stop>

each step applies a softmax over all vocab

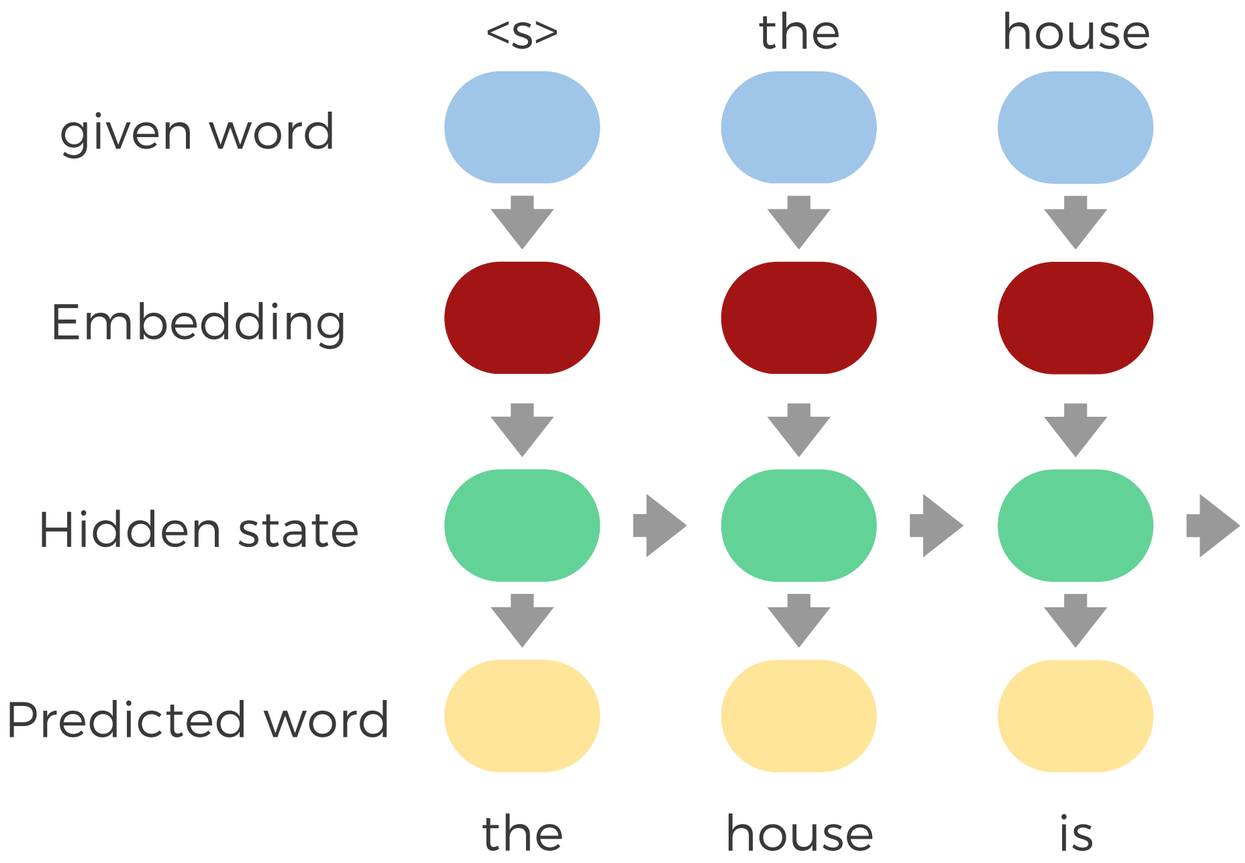
EXAMPLE



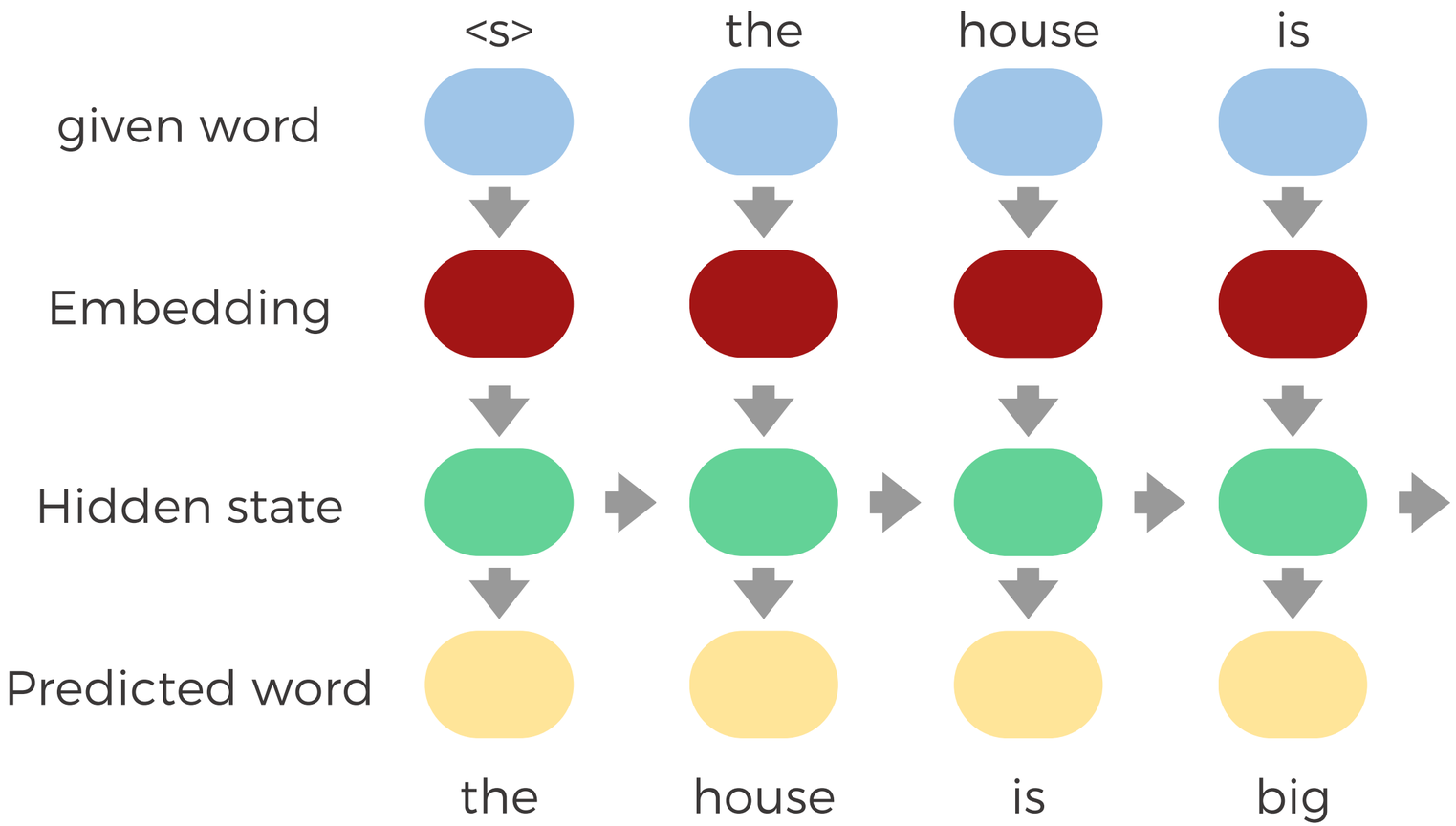
EXAMPLE



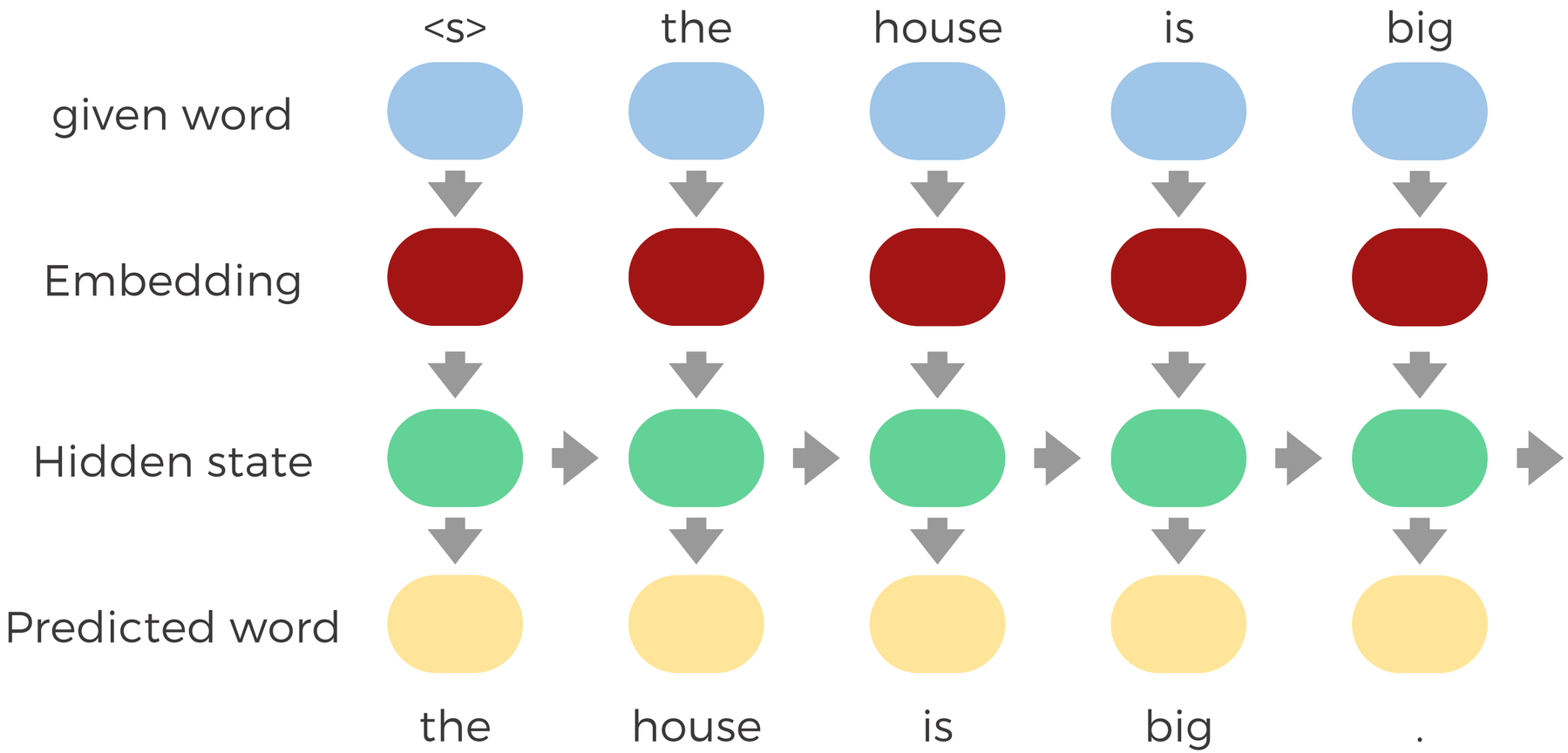
EXAMPLE



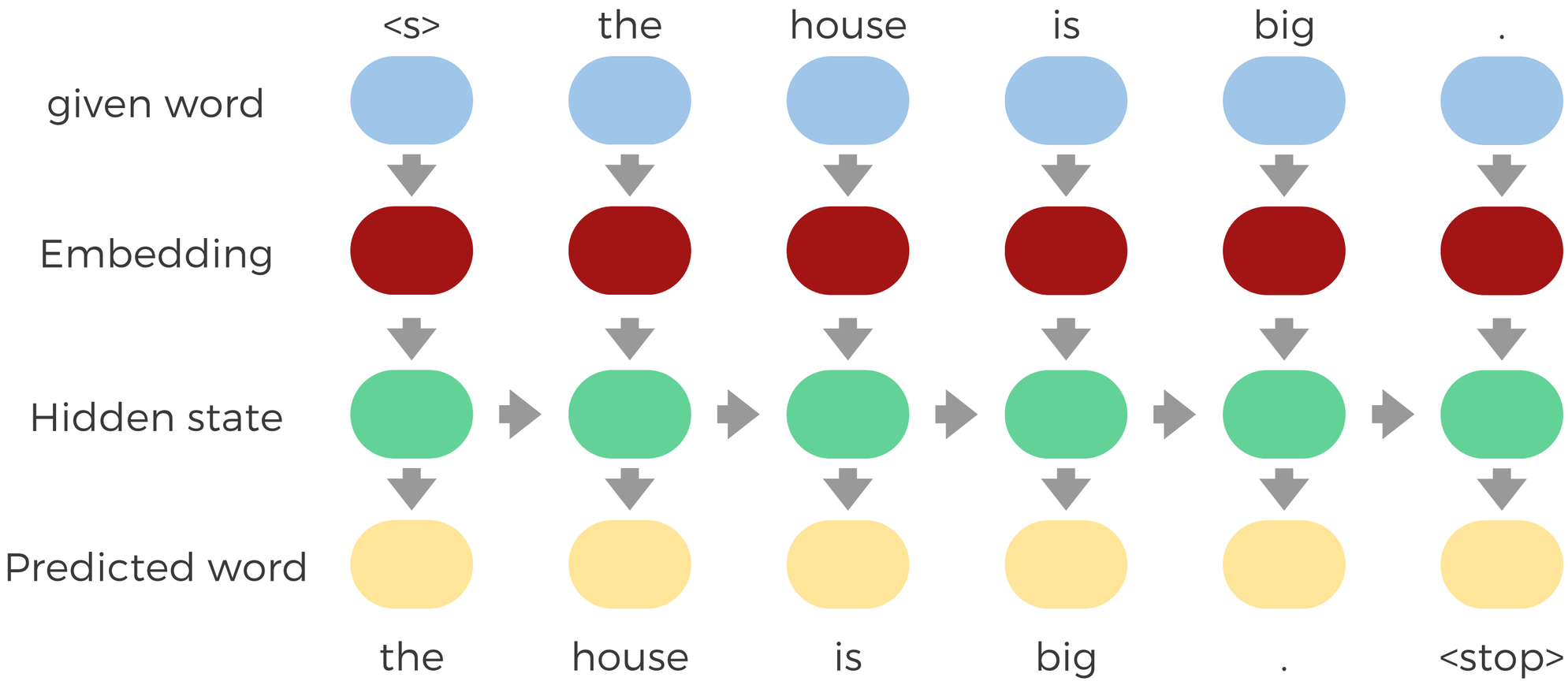
EXAMPLE


<s>
the
house
is
given word
Embedding
Hidden state
Predicted word
the
house
is
big

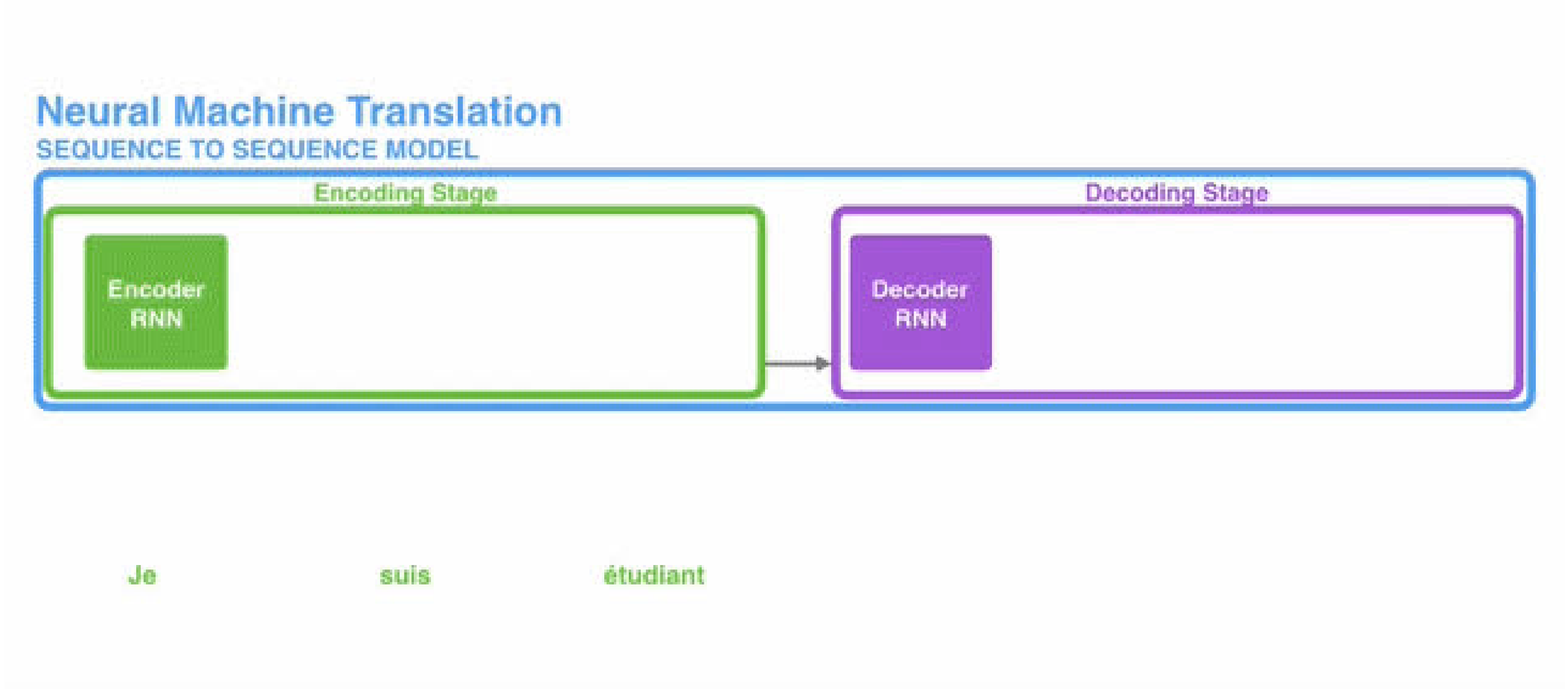
EXAMPLE



EXAMPLE



SEQ2SEQ ANIMATION



Attention Model

Indonesia AI

Transformers

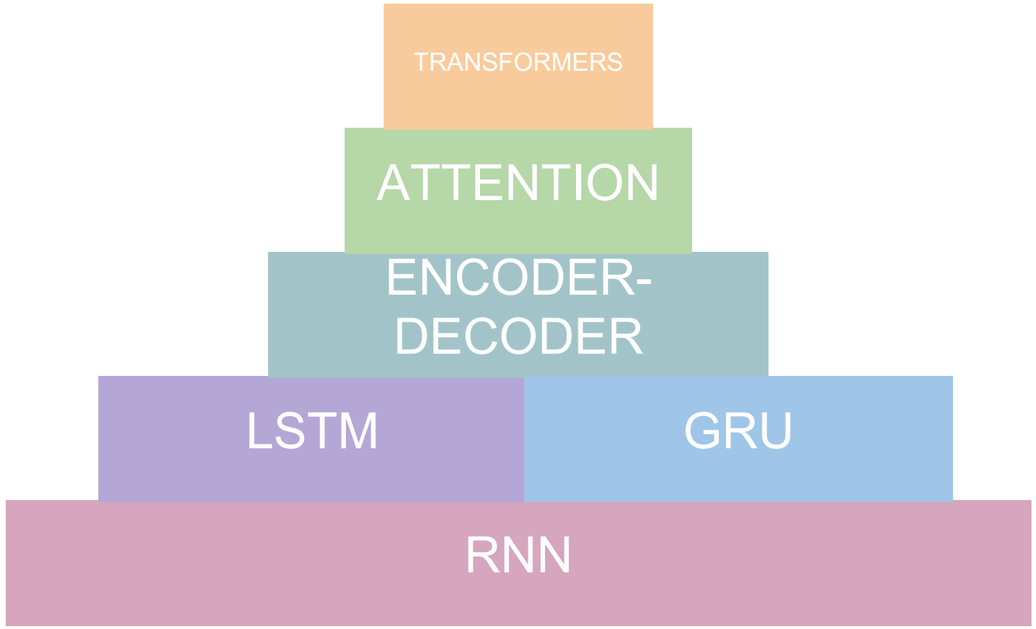
Paper

Attention Is All You Need
(2017)

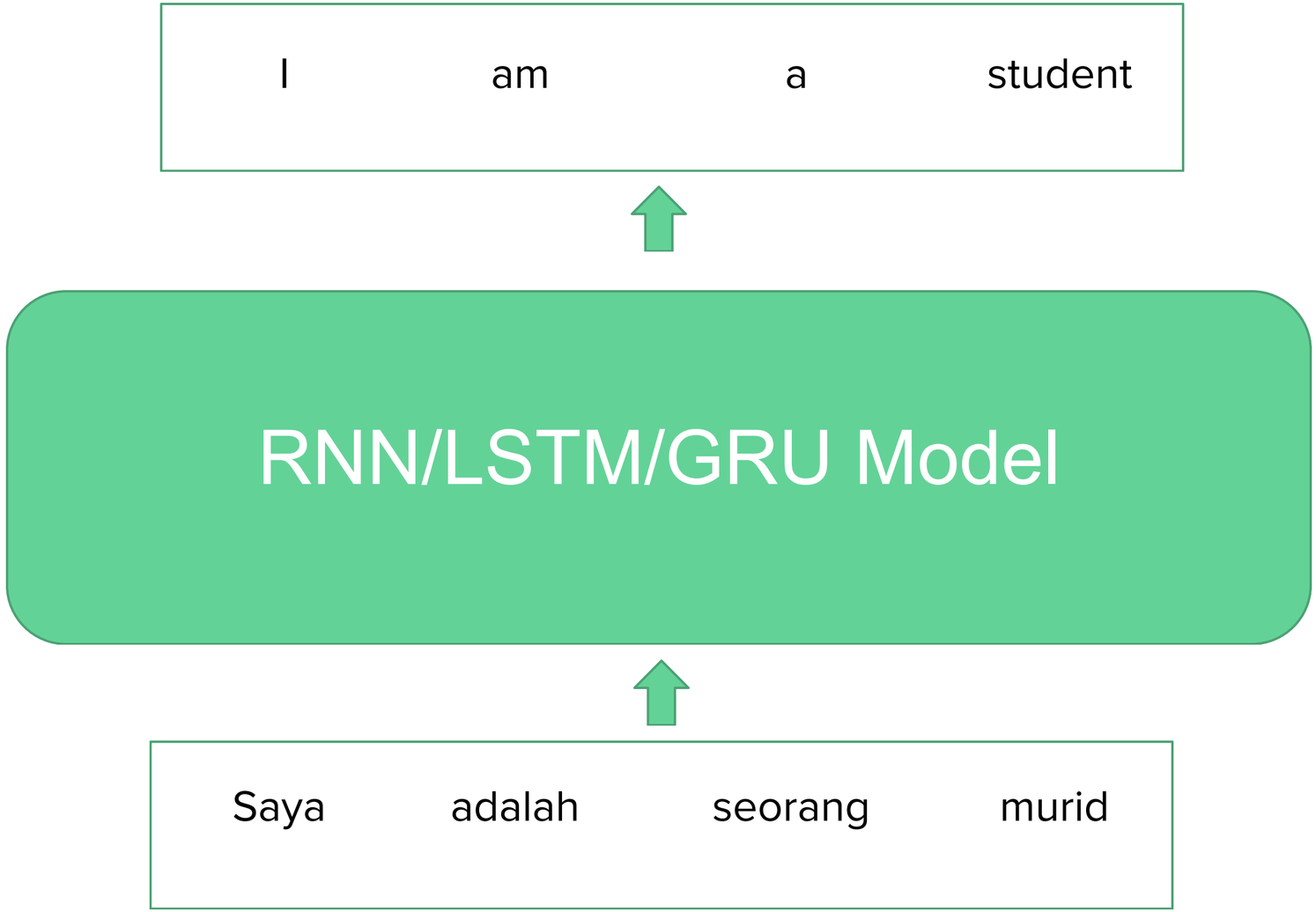
New family of neural network architecture

SoTA for Machine Translation, beats Google MT

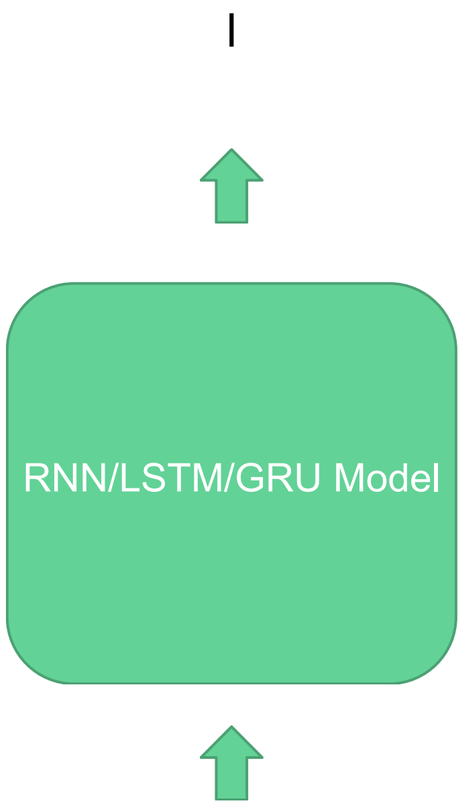
Designed for parallelization



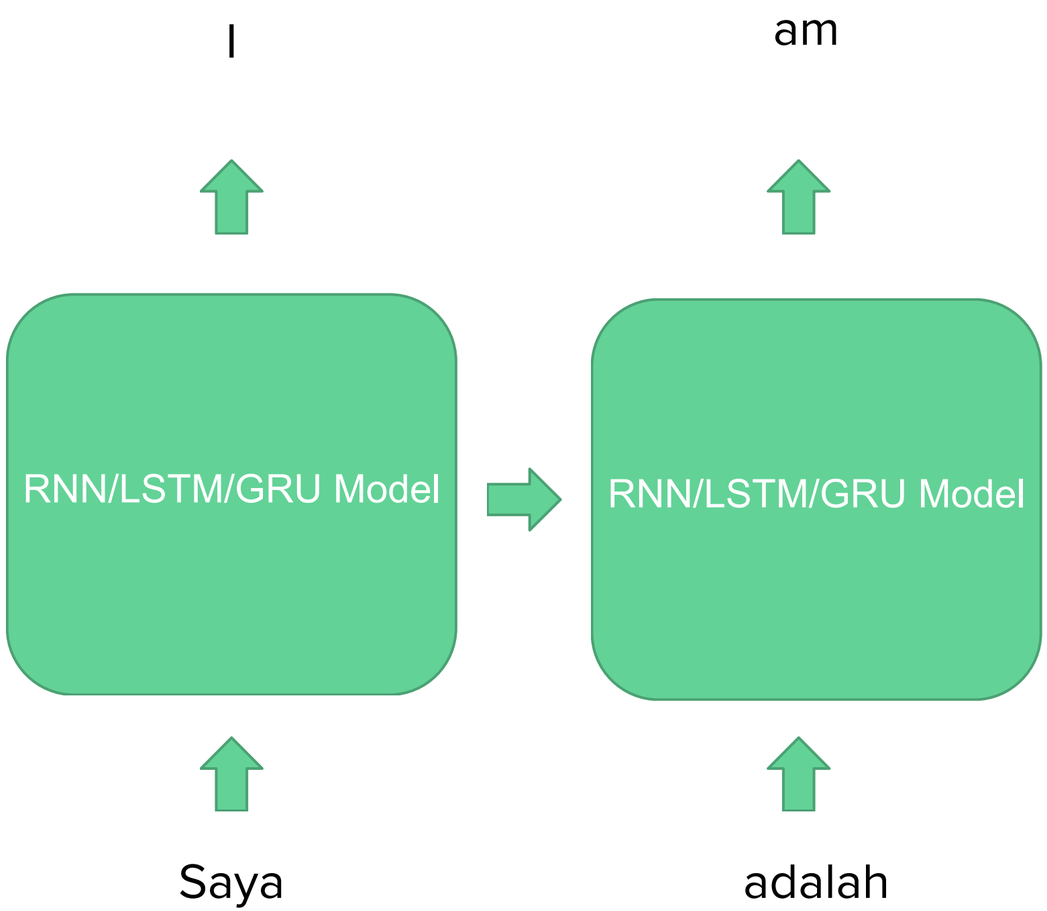
Old VS New Approach



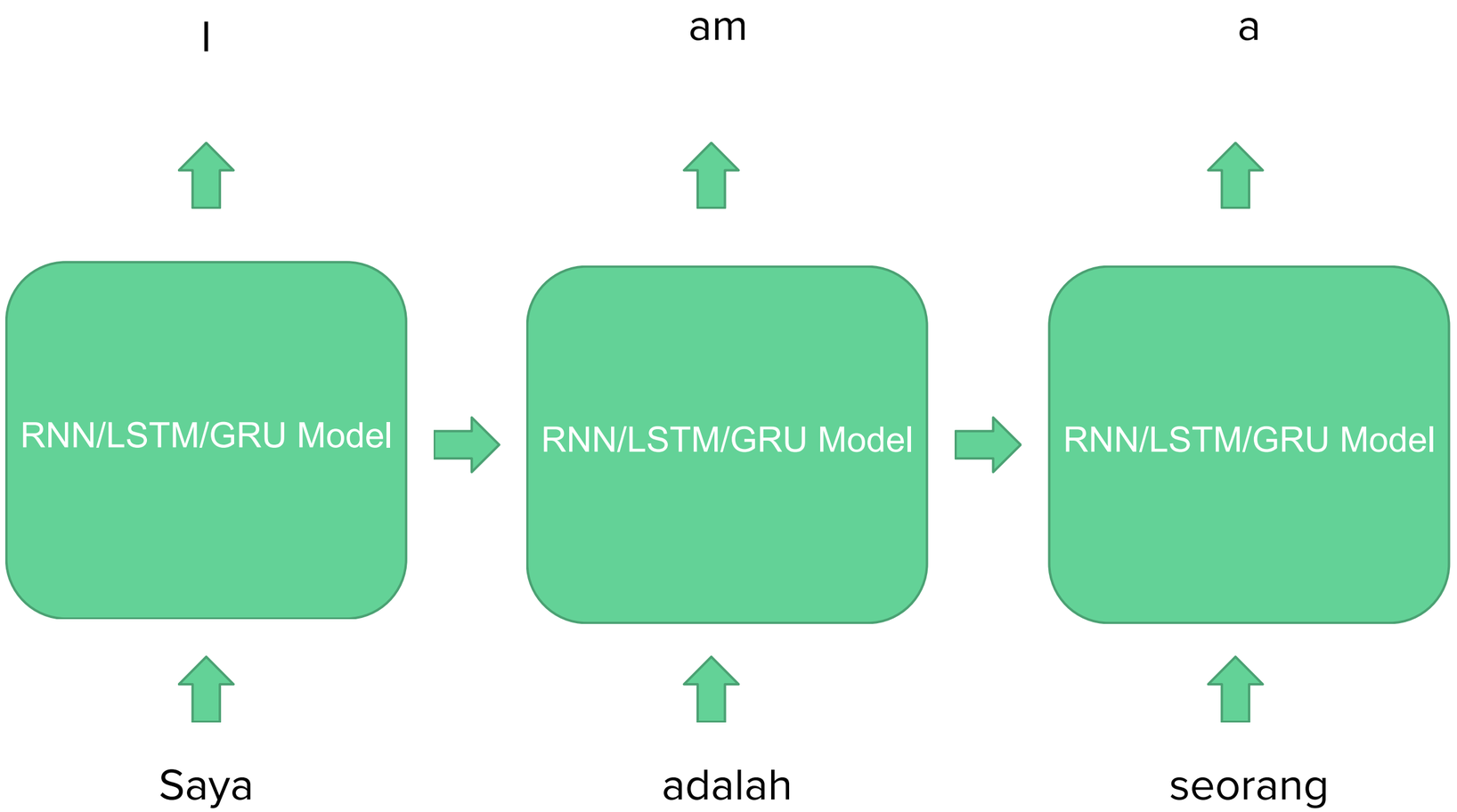
Old VS New Approach



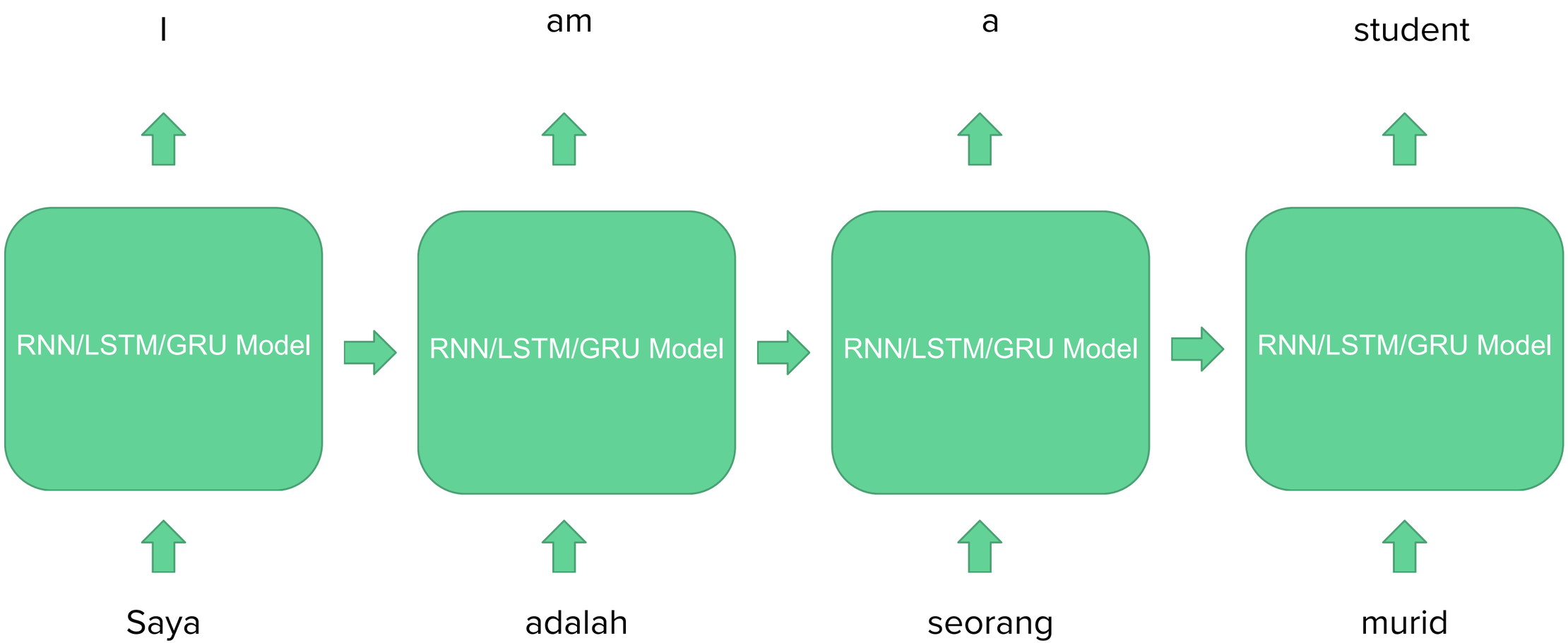
Old VS New Approach



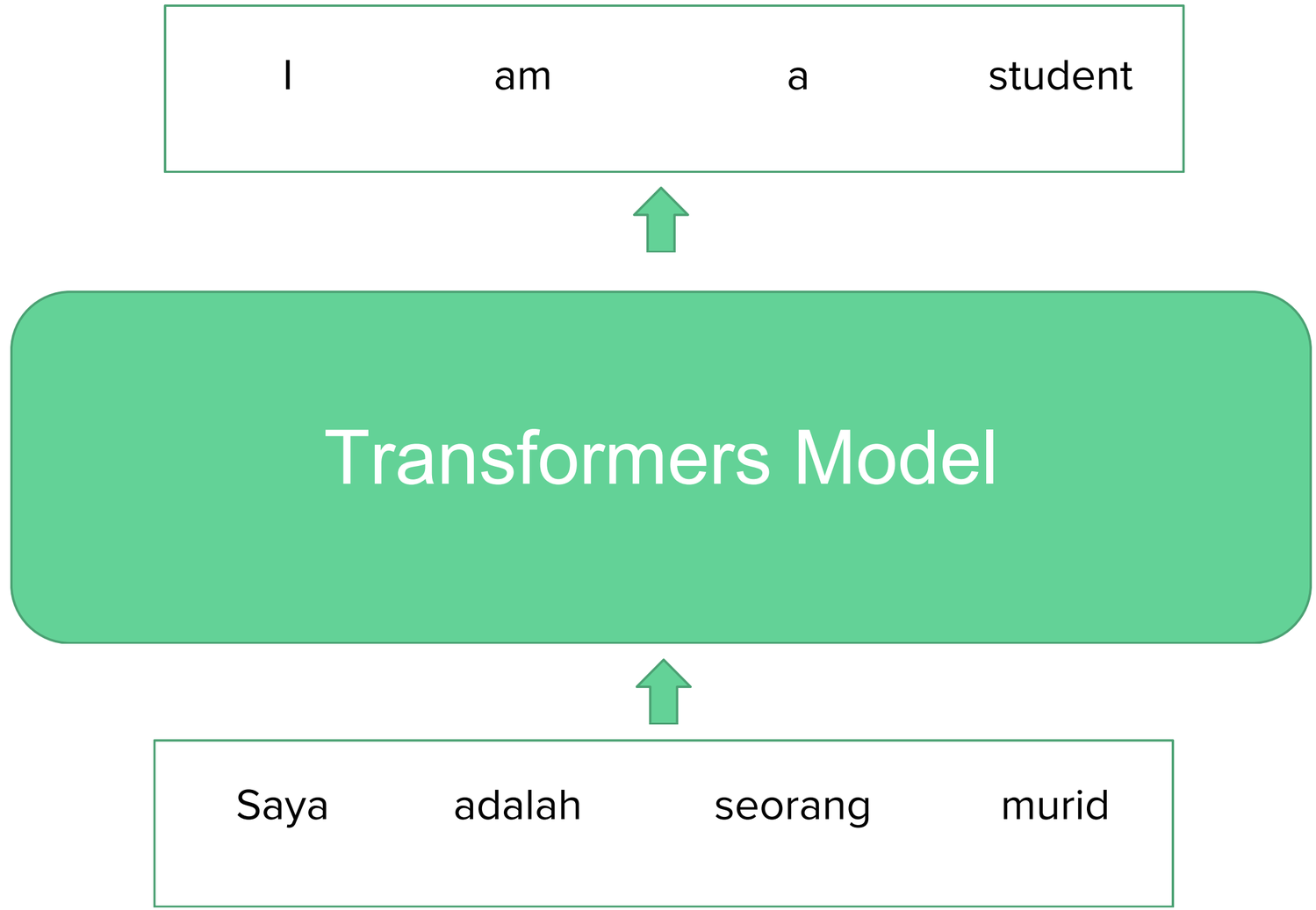
Old VS New Approach



Old VS New Approach



Old VS **New** Approach

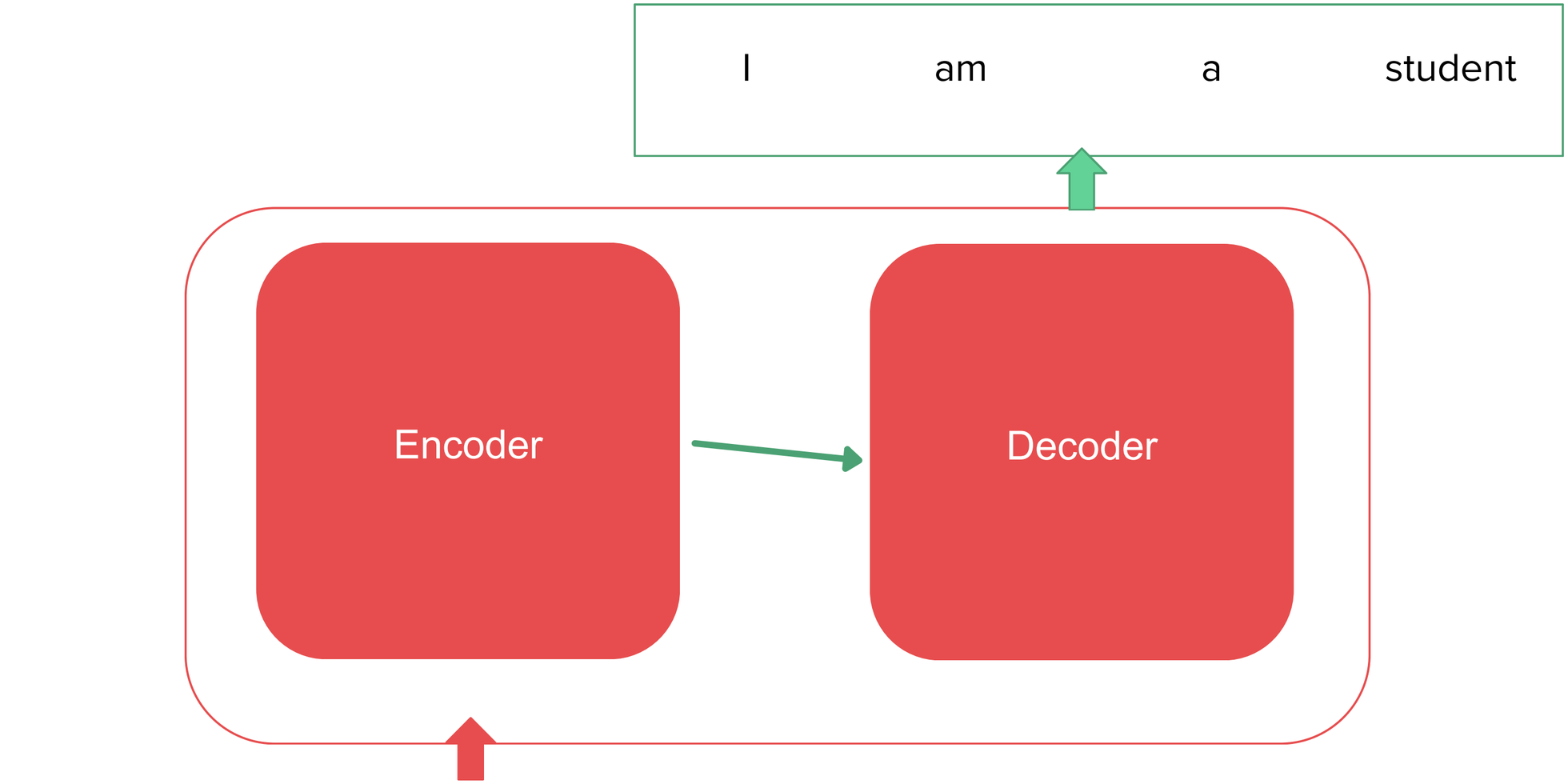


Any Question Guys ~

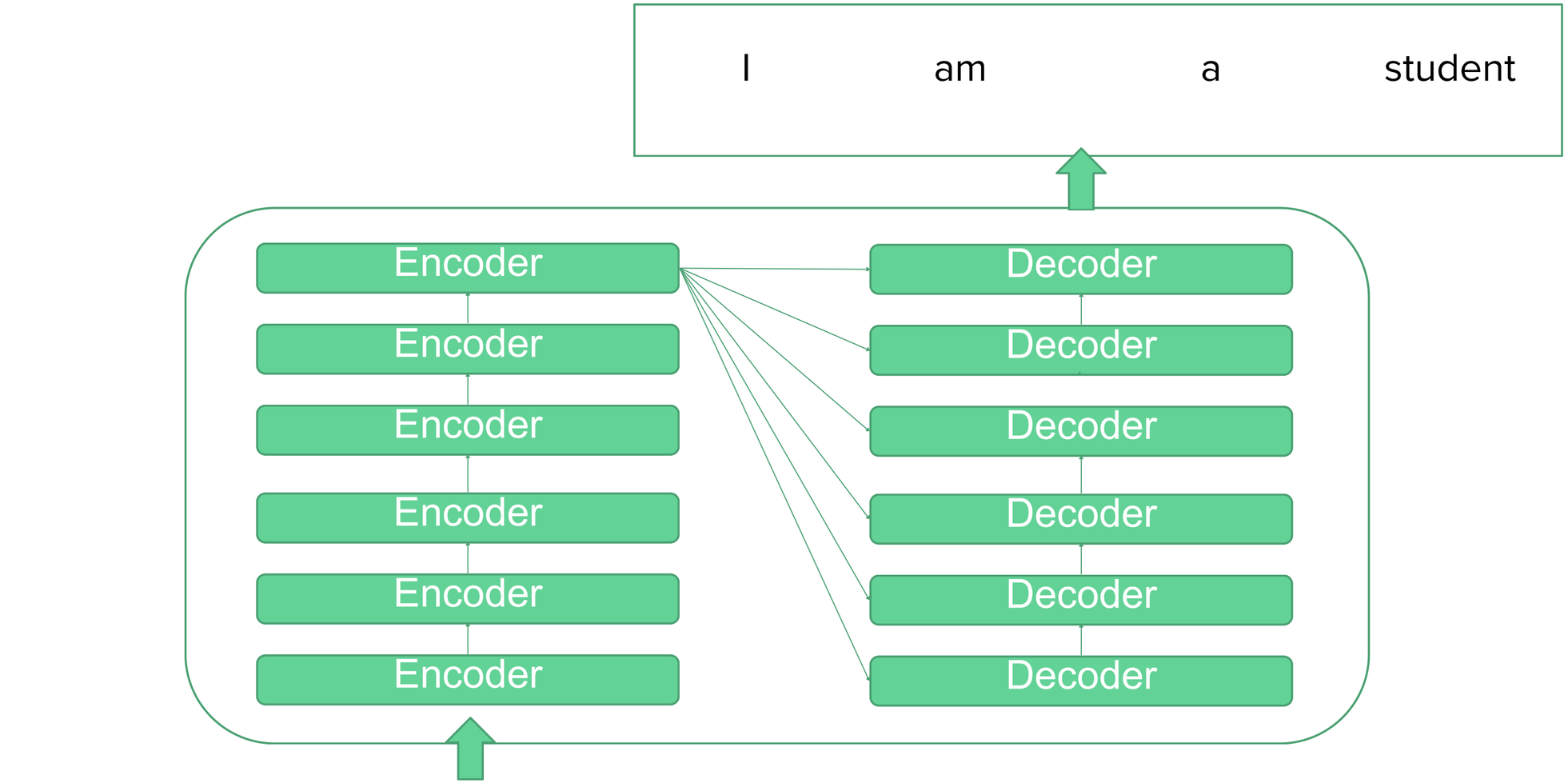
Inside The Transformers

Indonesia AI

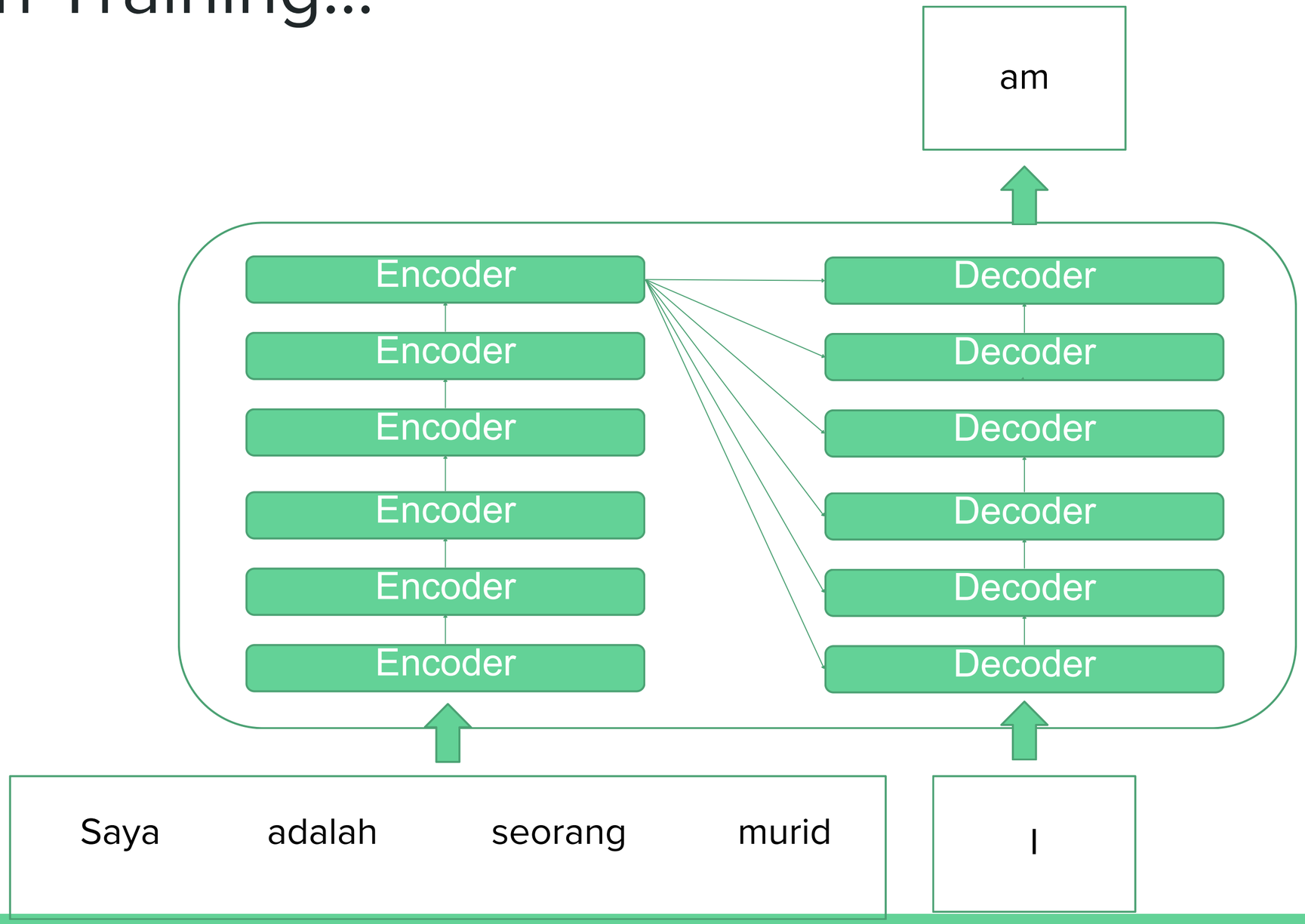
Inside Transformers



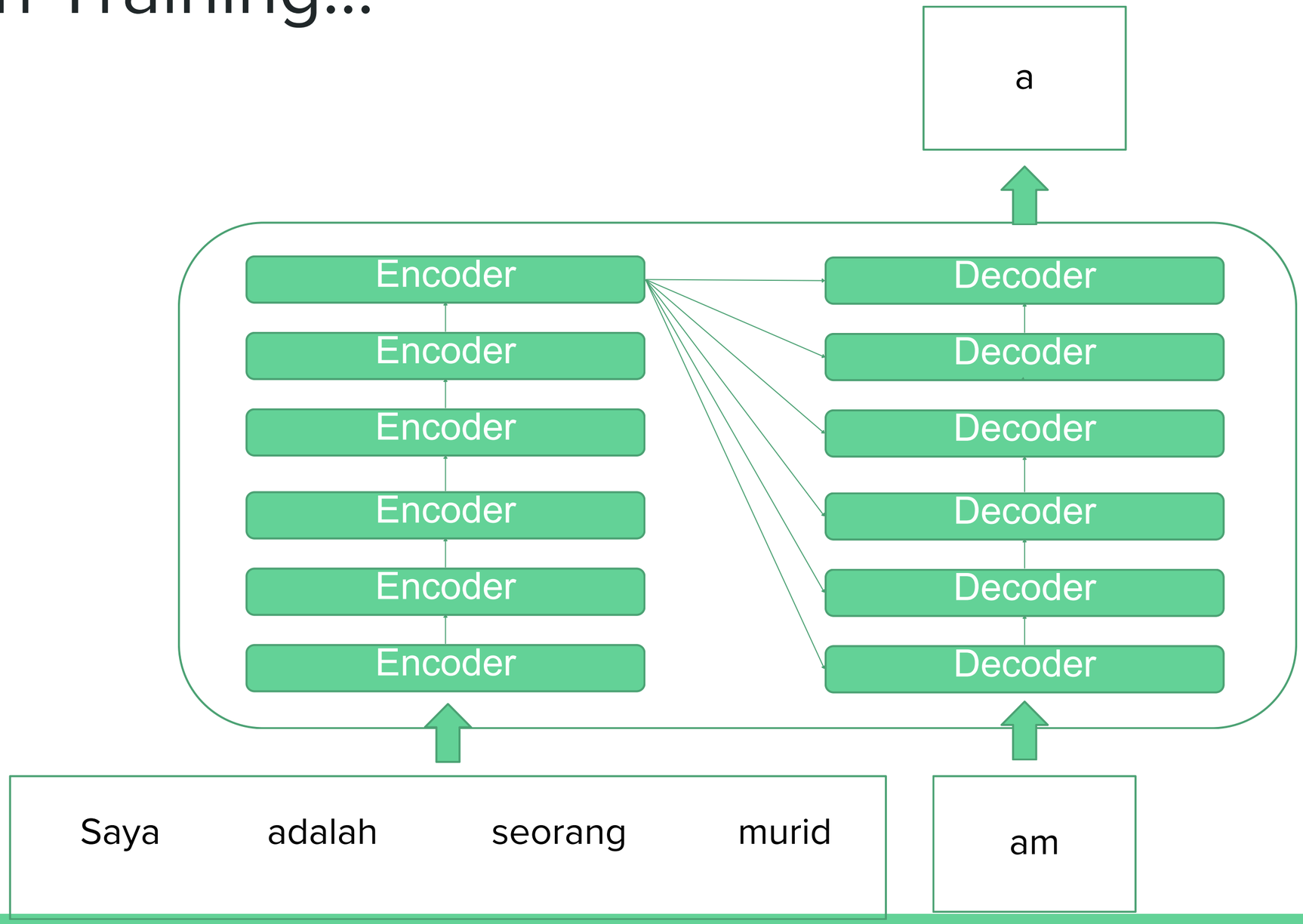
Inside Transformers



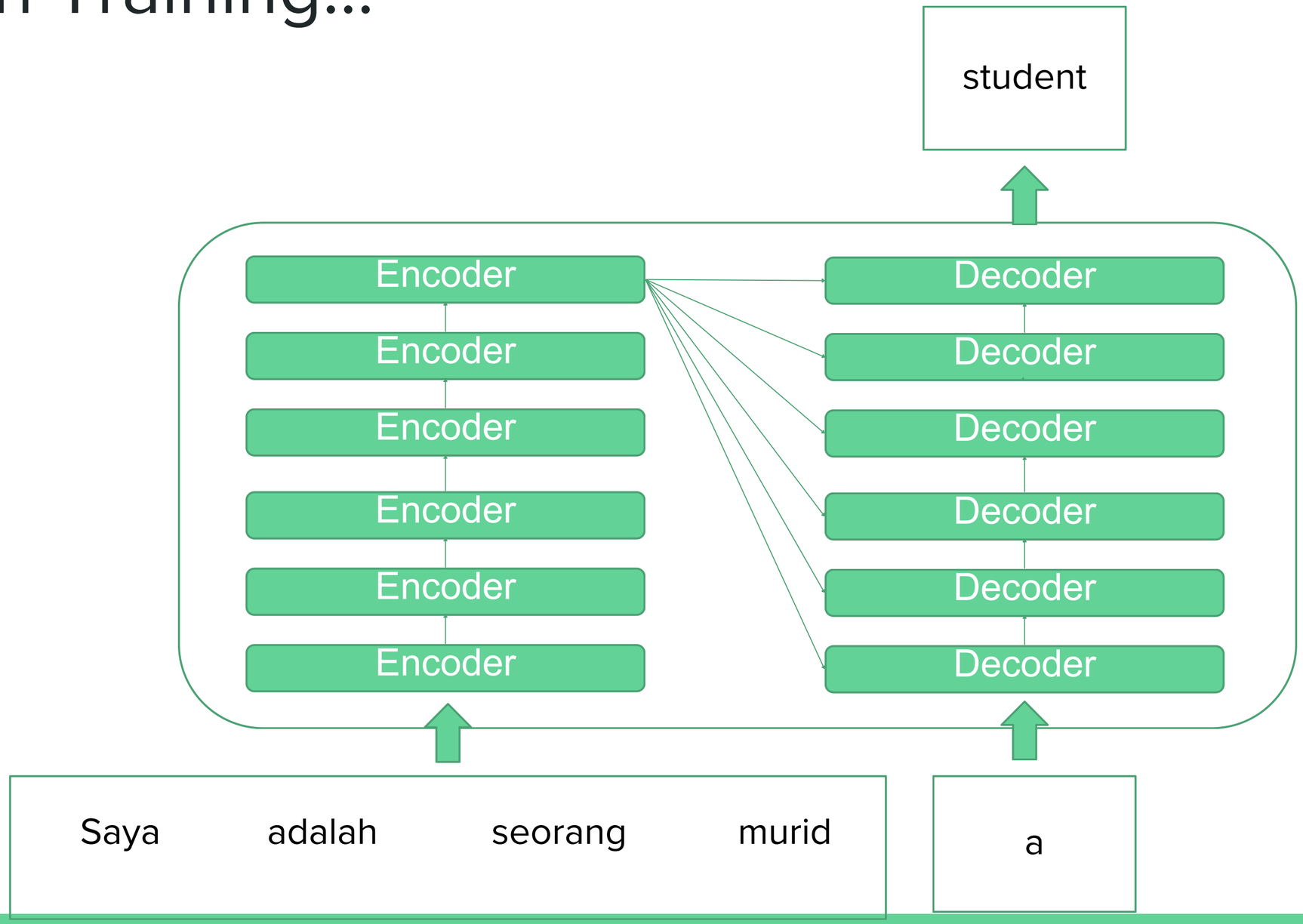
When Training...



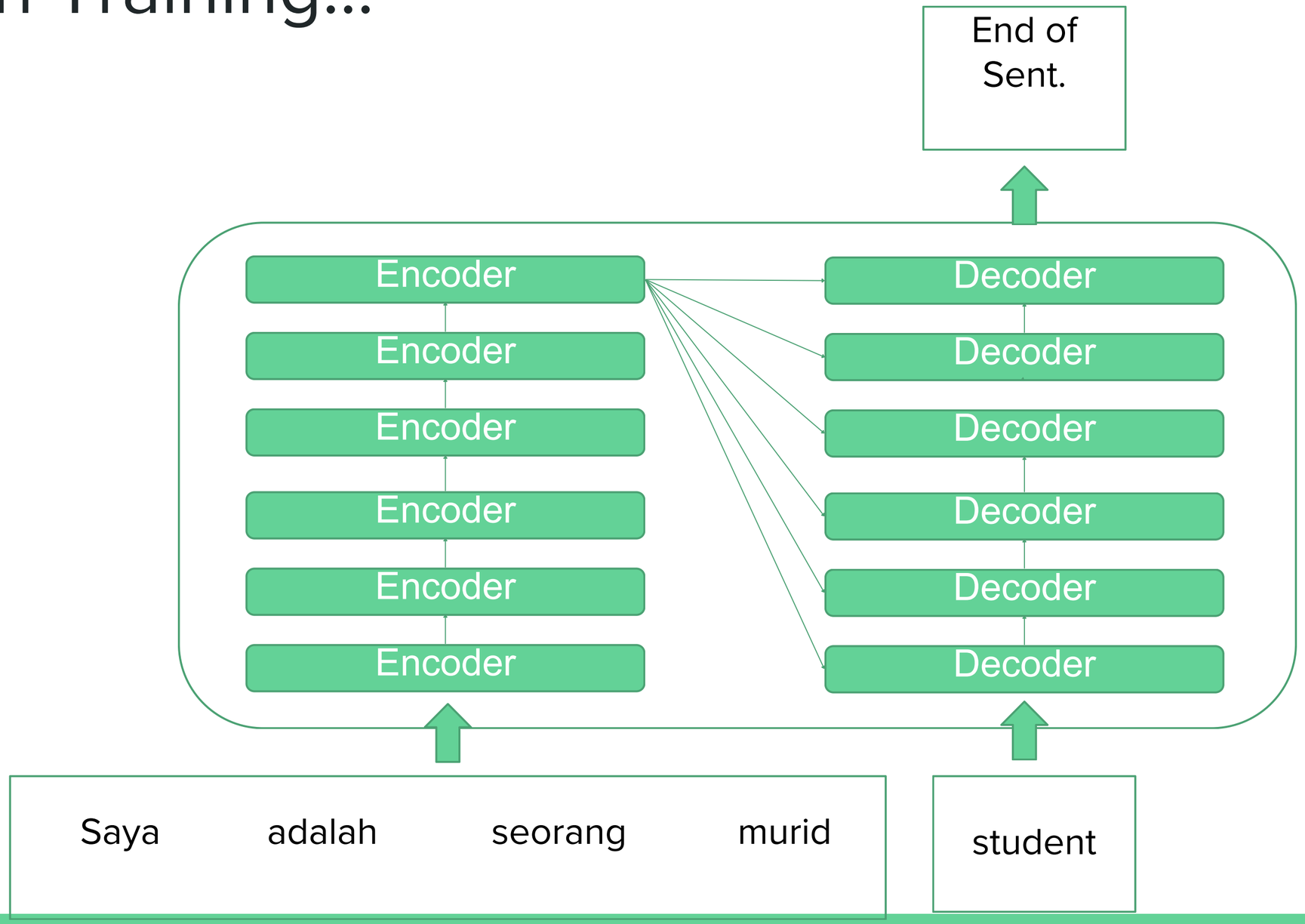
When Training...



When Training...

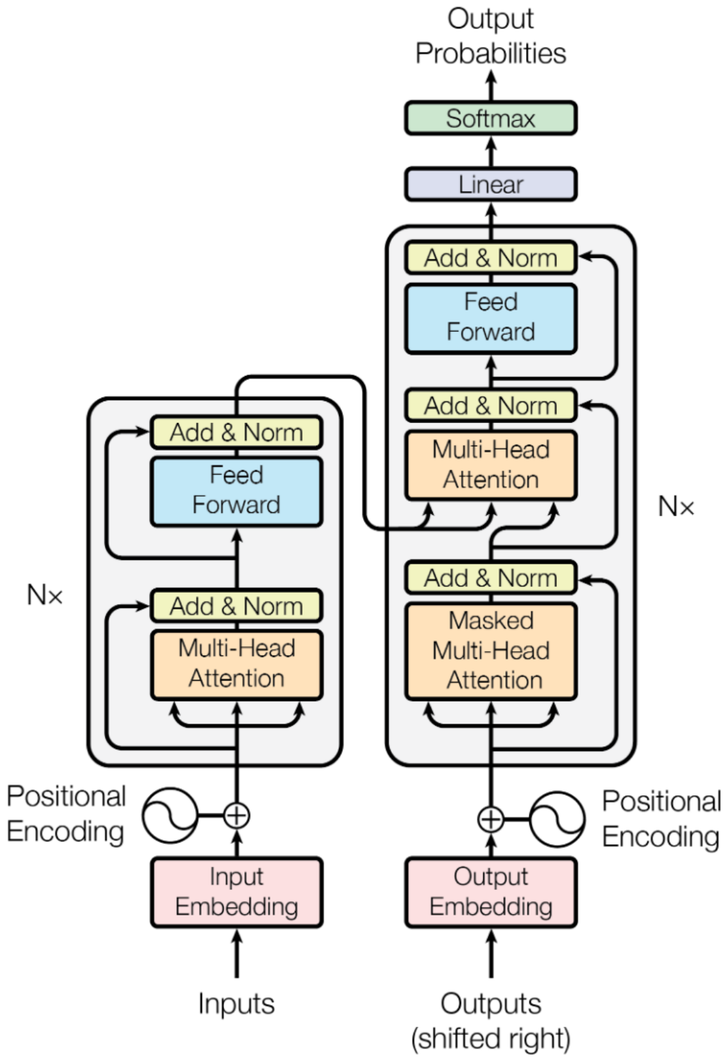


When Training...



Architecture

Encoder



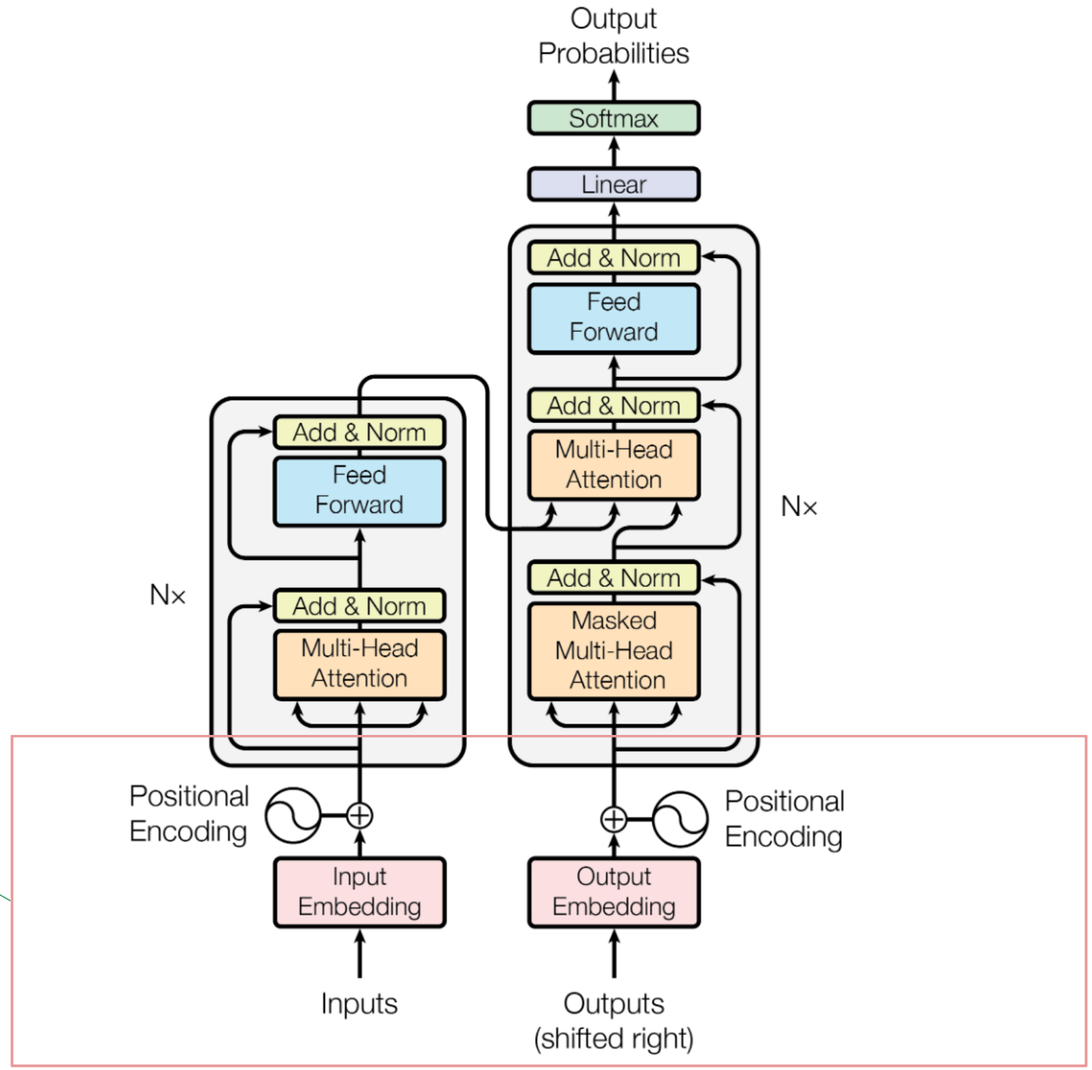
Decoder

Architecture

Encoder

Decoder

Let's talk about this first



Any Question Guys ~

Transformer's Positional Embedding

Indonesia AI

Positional Embedding

Purpose:

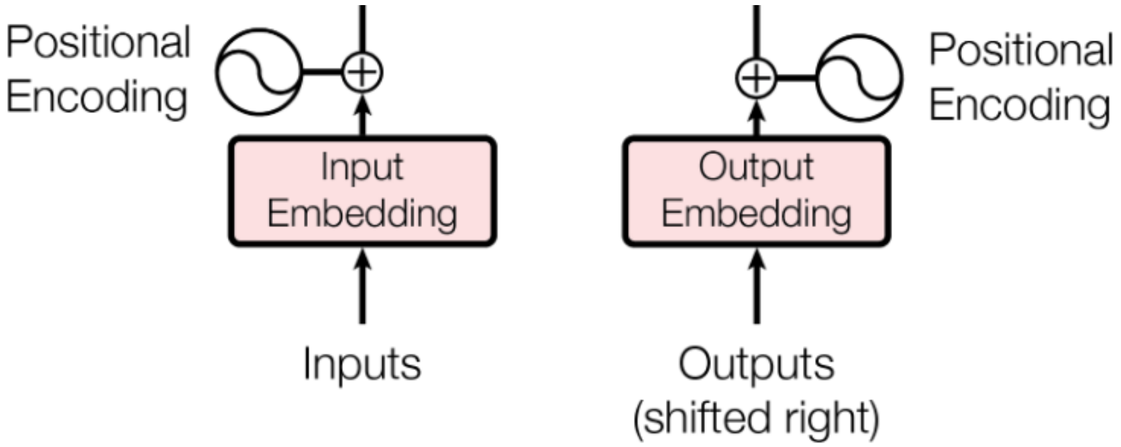
Understanding **context** of words based on the **position** in a sentence.

Input example:

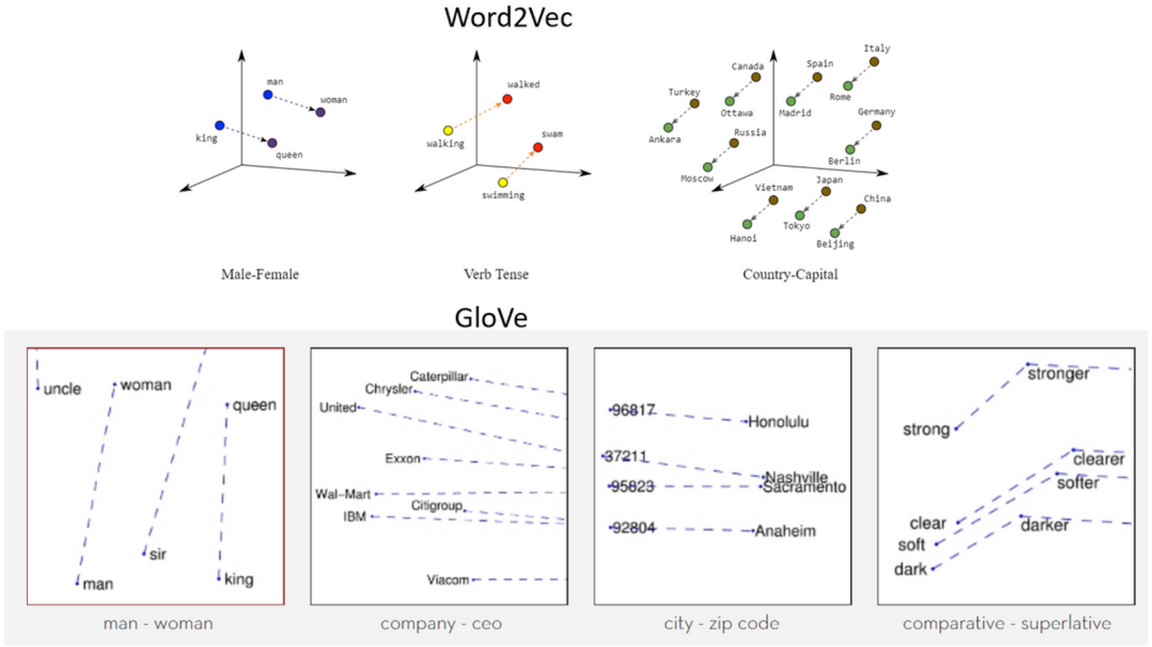
- Pak **guru** sedang mengajar
- Fhadli adalah seorang **guru**

Formula:

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



Positional Embedding



Word Embedding

Saya adalah seorang murid

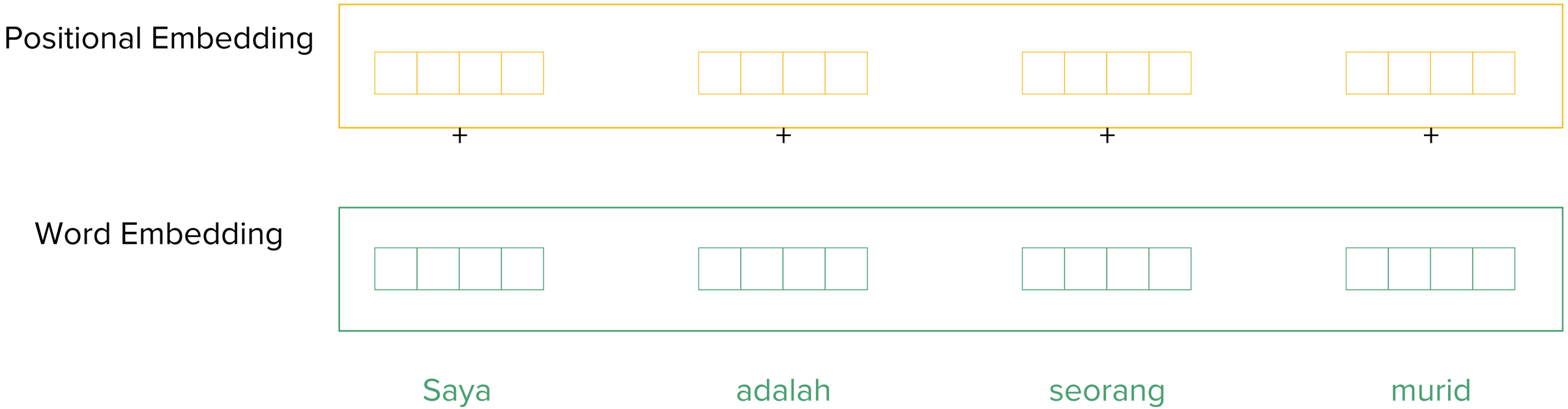
Positional Embedding

Can be changed

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

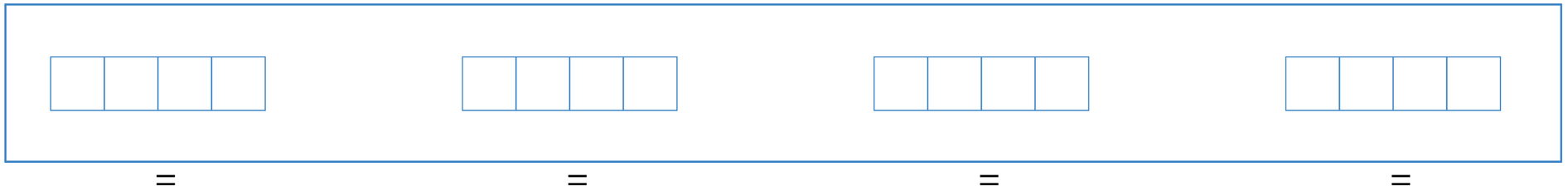
model will easily learn to attend by relative positions rather than only 0 & 1

Original paper use this equation to generate positional embedding



Positional Embedding

Real Embedding for Input



=

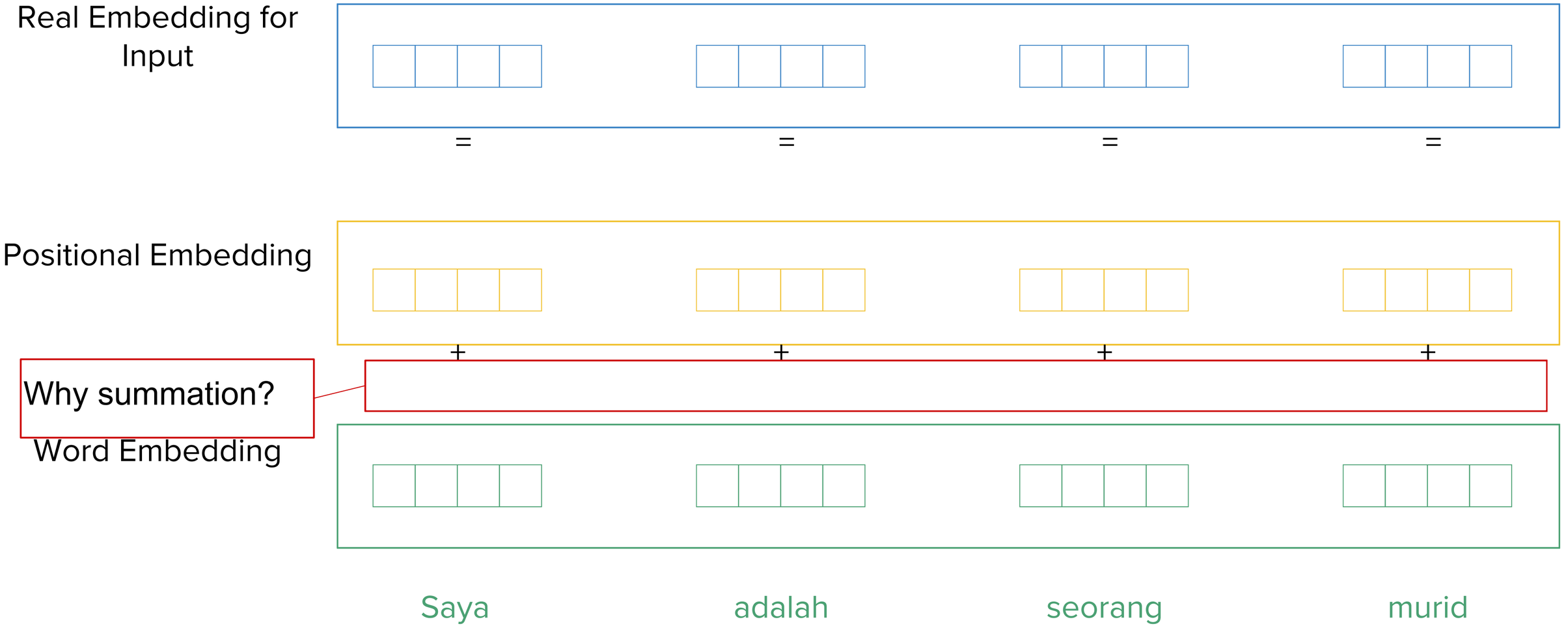
Positional Embedding

+

Word Embedding

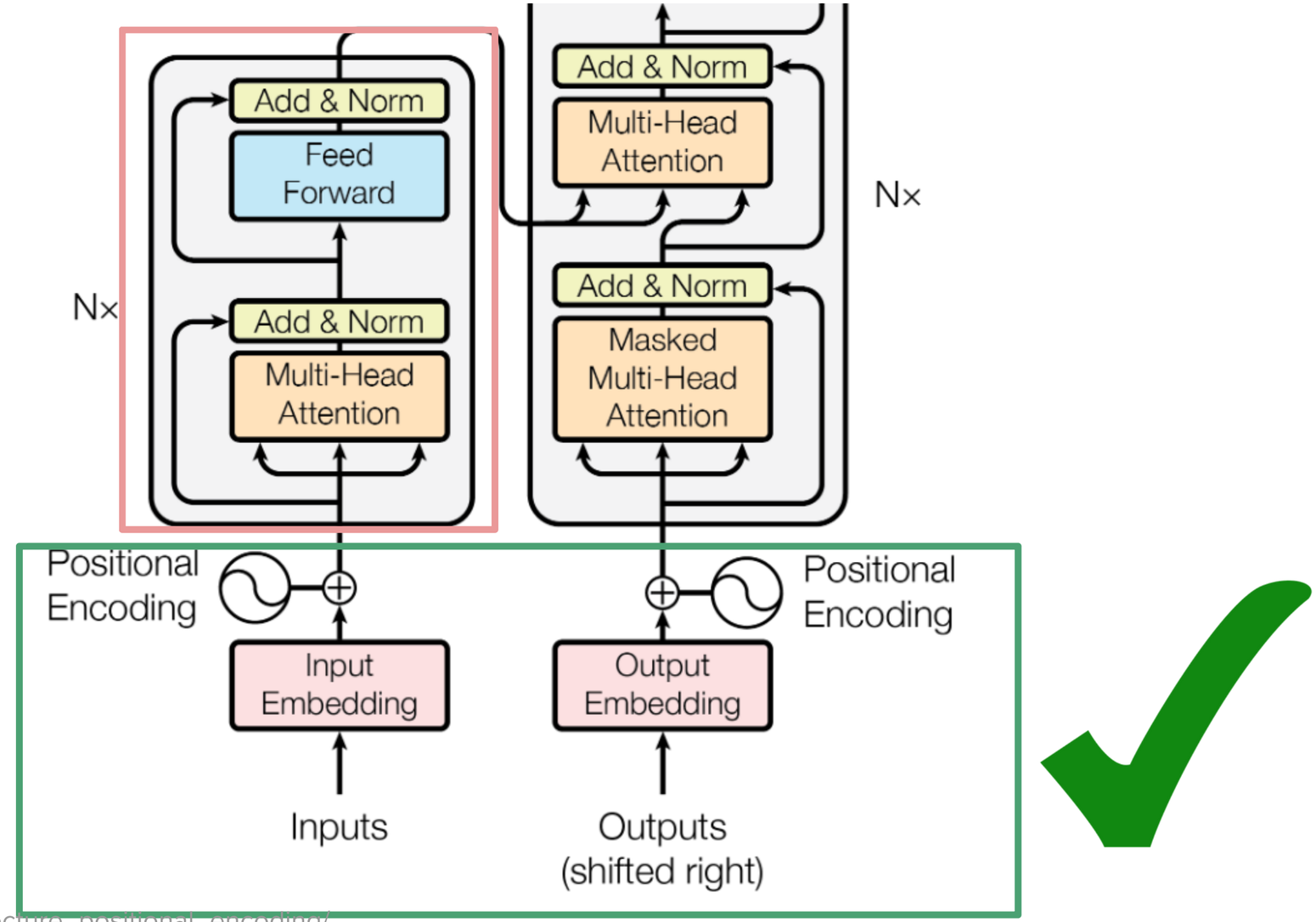
Saya adalah seorang murid

Positional Embedding



Positional Embedding

Let's talk about the encoder



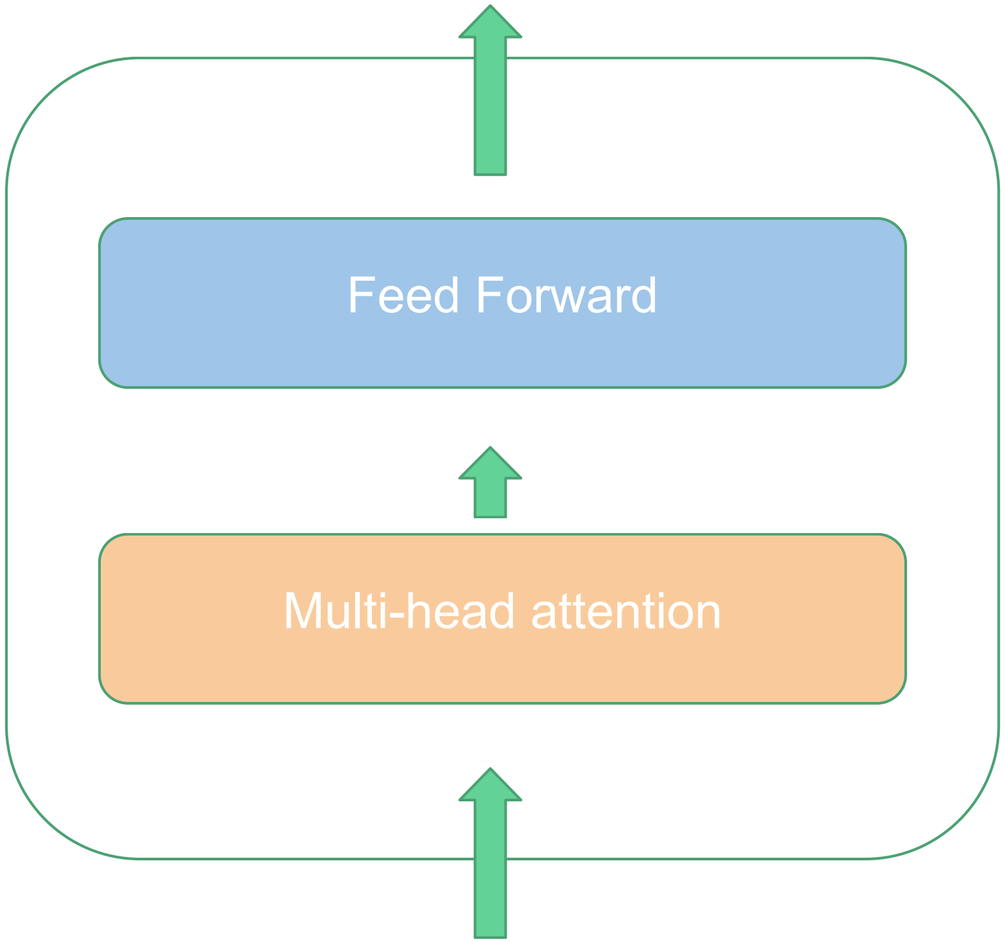
https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Any Question Guys ~

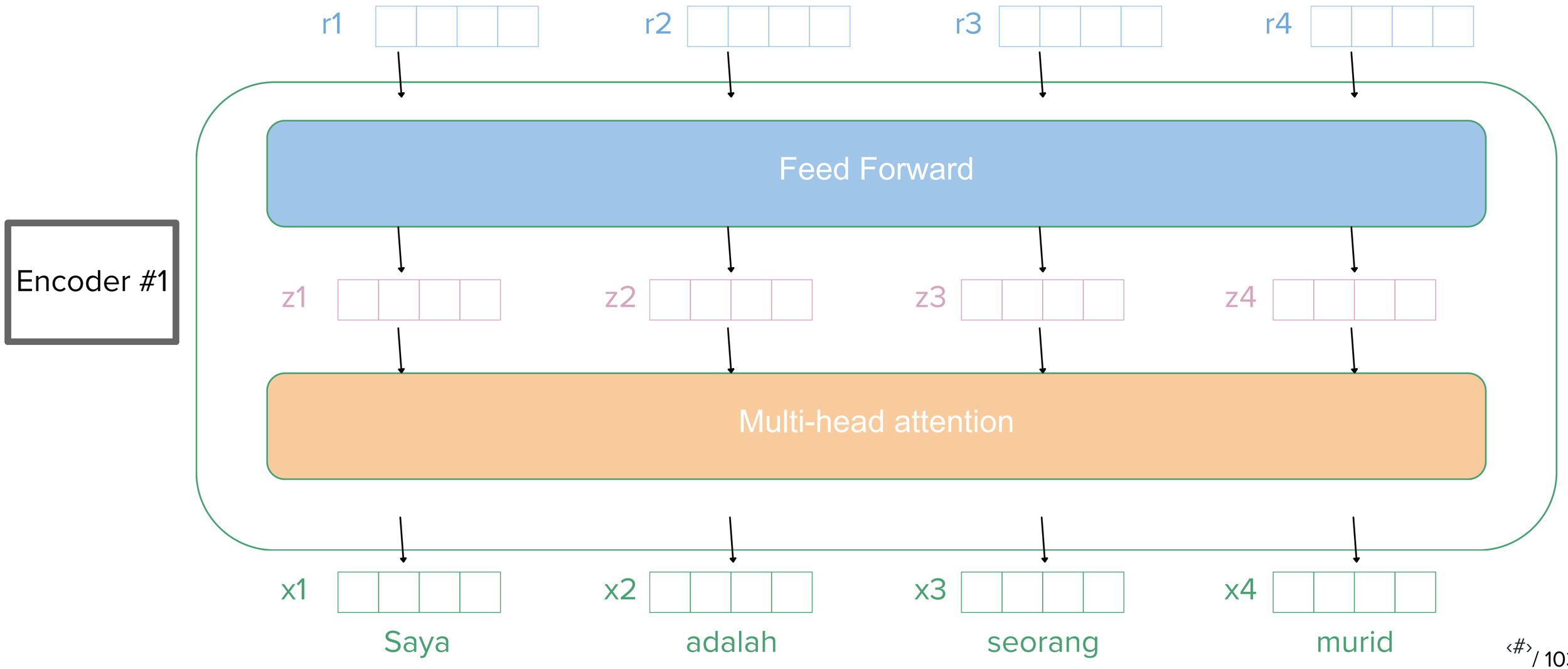
Transformer's Encoder

Indonesia AI

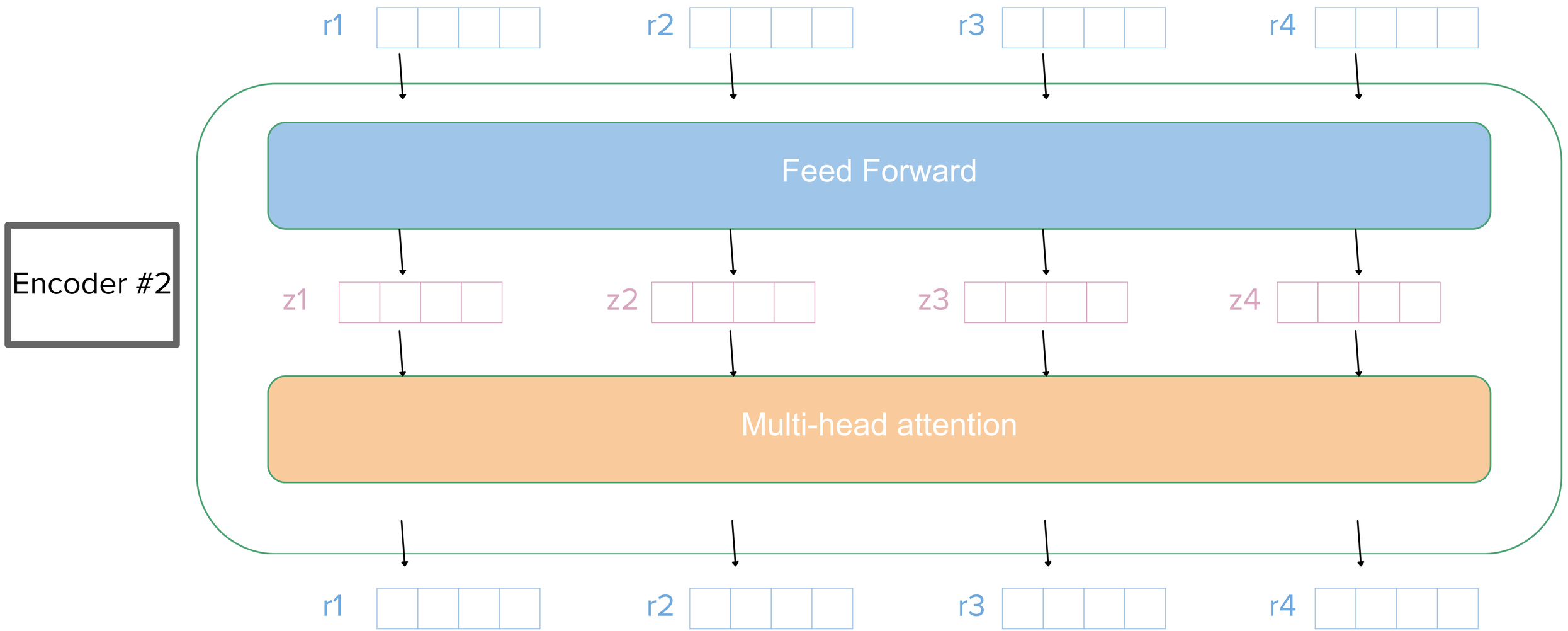
Encoder



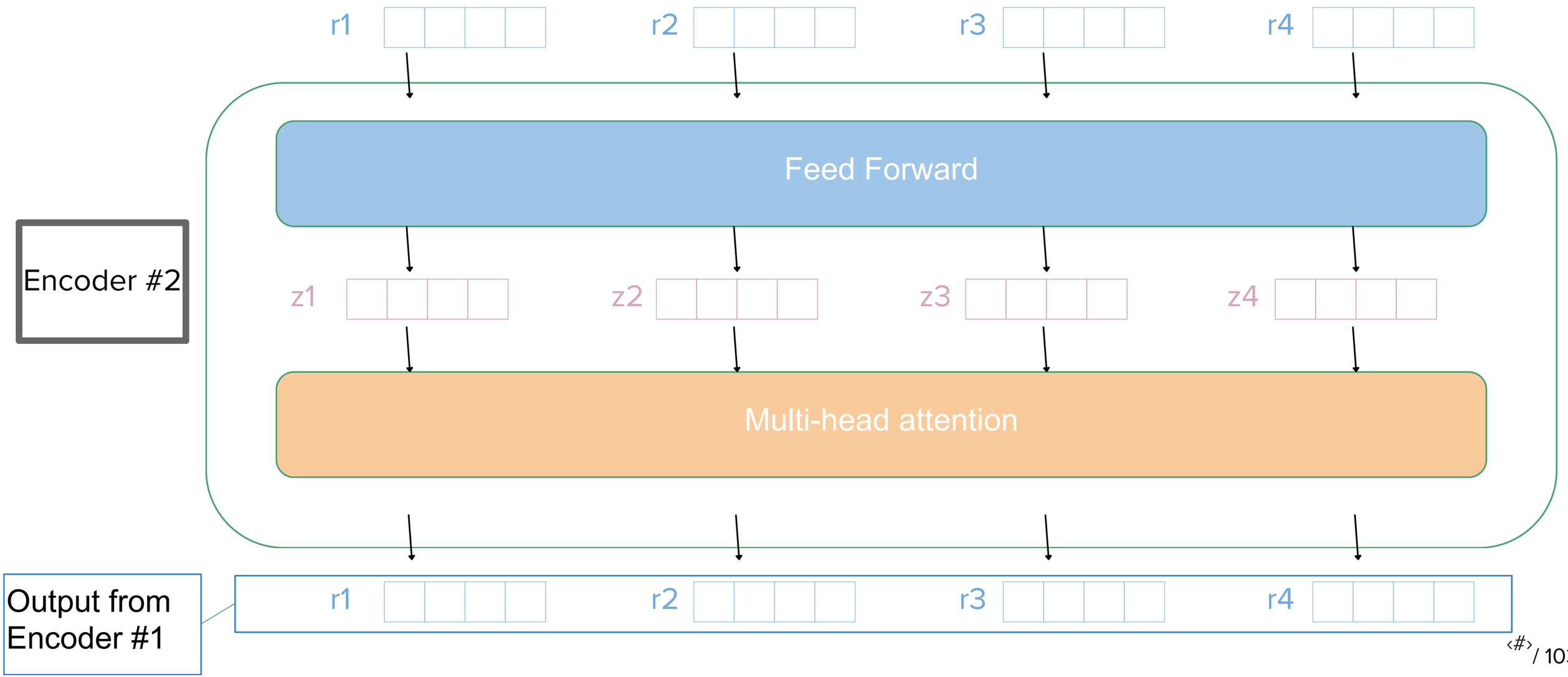
Encoder


r1
r2
r3
r4
Feed Forward
Encoder #1
z1
z2
z3
z4
Multi-head attention
x1
x2
x3
x4
Saya
adalah
seorang
murid

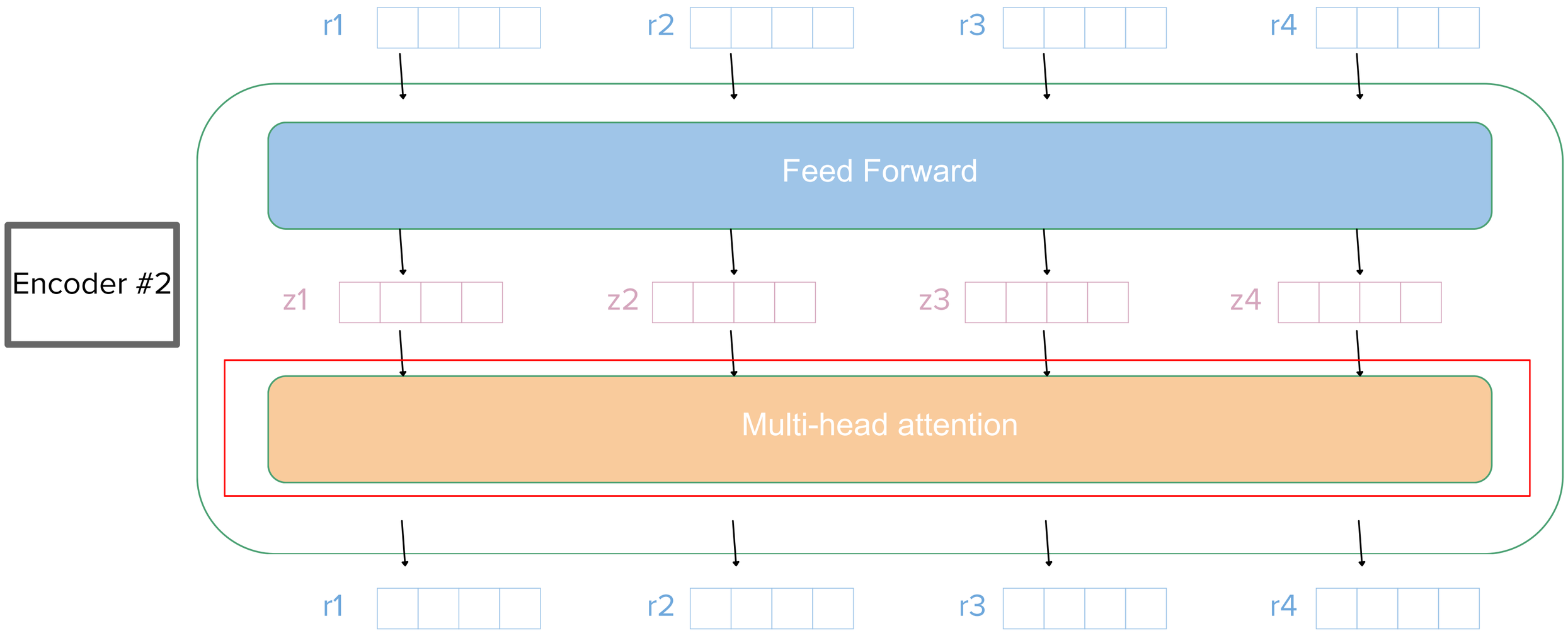
Encoder


r1
r2
r3
r4
Feed Forward
Encoder #2
z1
z2
z3
z4
Multi-head attention
r1
r2
r3
r4

Encoder


r1
r2
r3
r4
Feed Forward
Encoder #2
z1
z2
z3
z4
Multi-head attention
Output from Encoder #1
r1
r2
r3
r4

Encoder


r1
r2
r3
r4
Feed Forward
Encoder #2
z1
z2
z3
z4
Multi-head attention
r1
r2
r3
r4

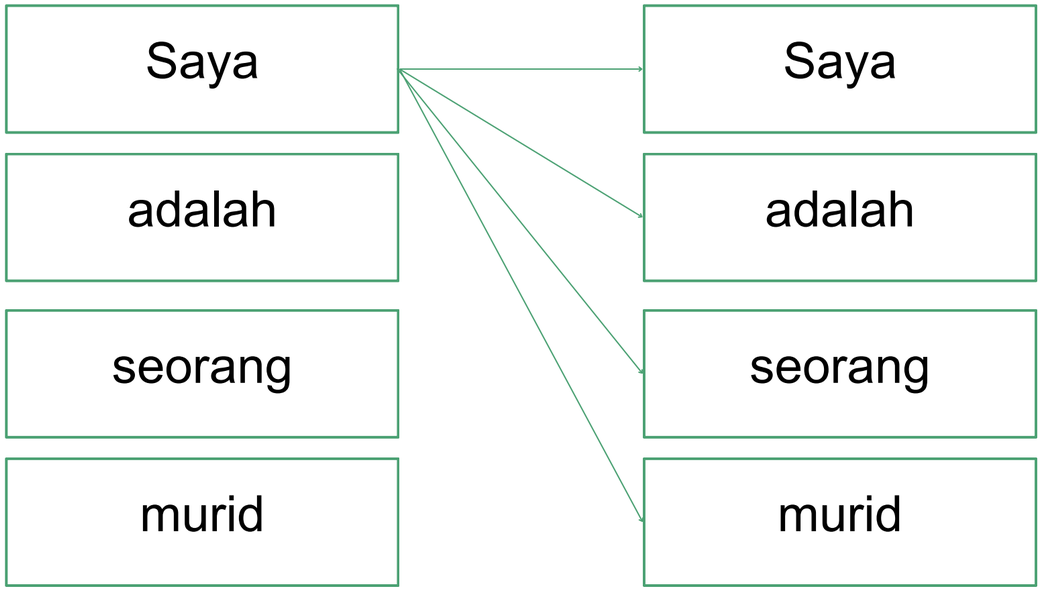
Encoder: Multi-head attention

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words in a sentence

   Example:

   Ari melihat pak **guru** sedang bersama anak **beliau**



It looks into the relationship of a word in a **sentence** and every words in that **sentence itself**. That's why it's also called **self-attention**

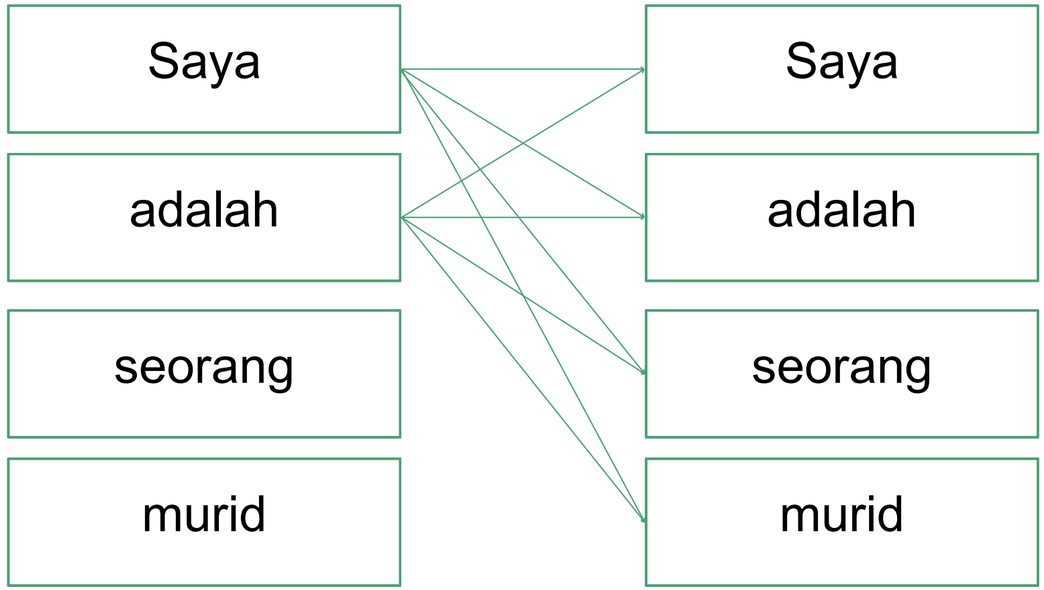
Encoder: Multi-head attention

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words in a sentence

   Example

   Ari melihat pak **guru** sedang bersama anak **beliau**



This is done between every words in the sentence. That's why it's also called **multi-headed self-attention**

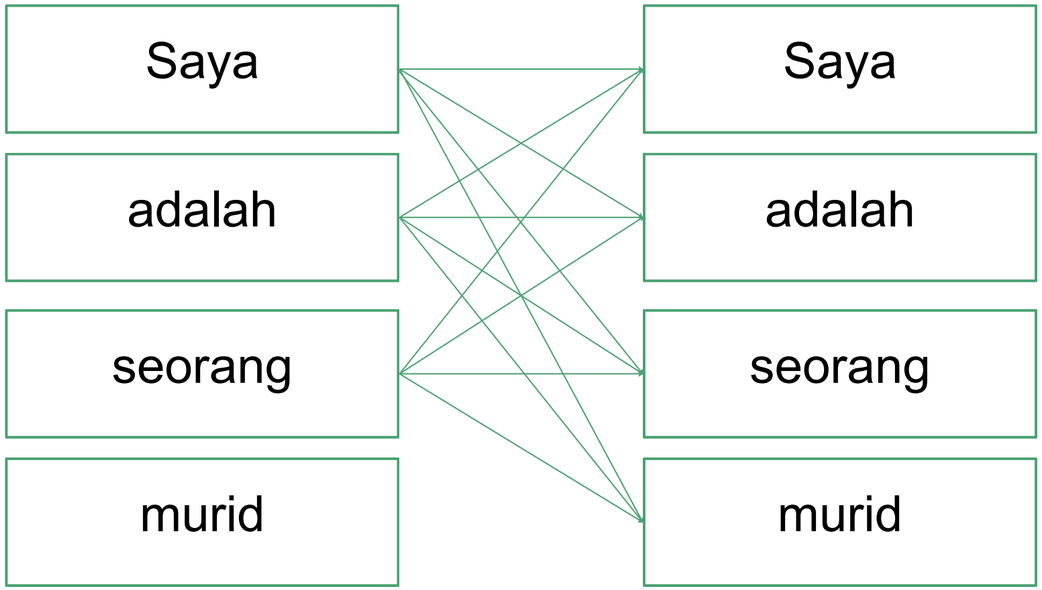
Encoder: Multi-head attention

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words in a sentence

   Example

   Ari melihat pak **guru** sedang bersama anak **beliau**



This is done between every words in the sentence. That's why it's also called **multi-headed self-attention**

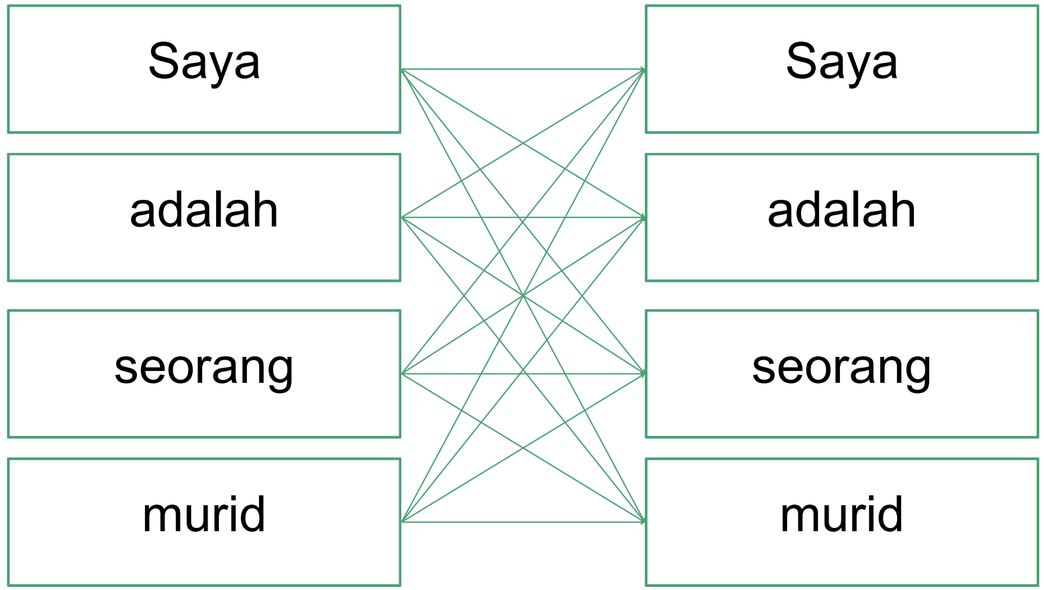
Encoder: Multi-head attention

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words in a sentence

   Example

   Ari melihat pak **guru** sedang bersama anak **beliau**



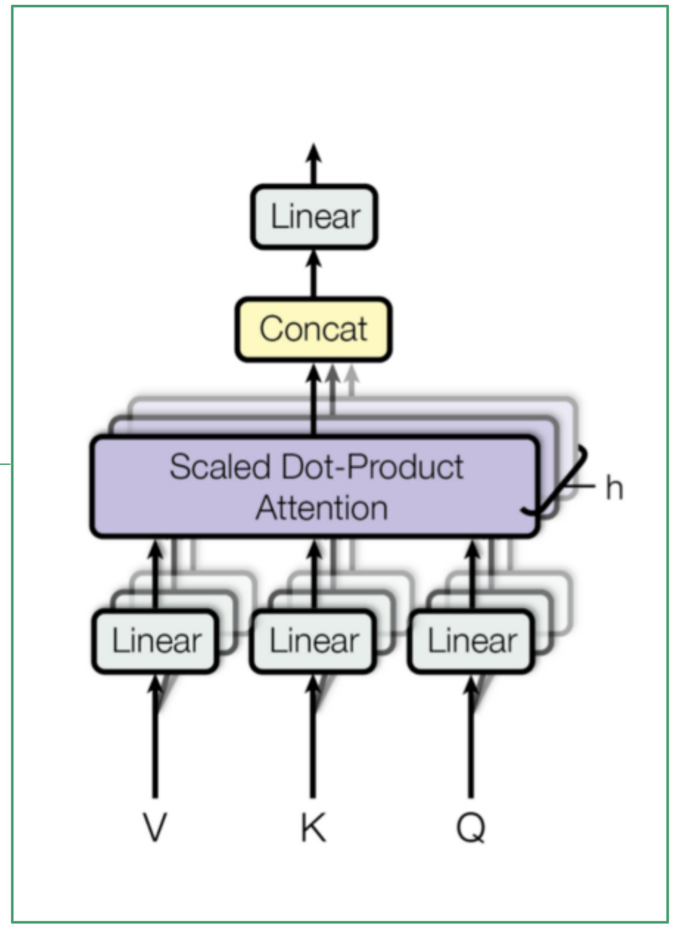
This is done between every words in the sentence. That's why it's also called **multi-headed self-attention**

Encoder: Multi-head attention

Saya	[0.73 0.04 0.05 0.18]
adalah	[0.03 0.84 0.02 0.11]
seorang	[0.10 0.06 0.62 0.22]
murid	[0.05 0.03 0.04 0.88]

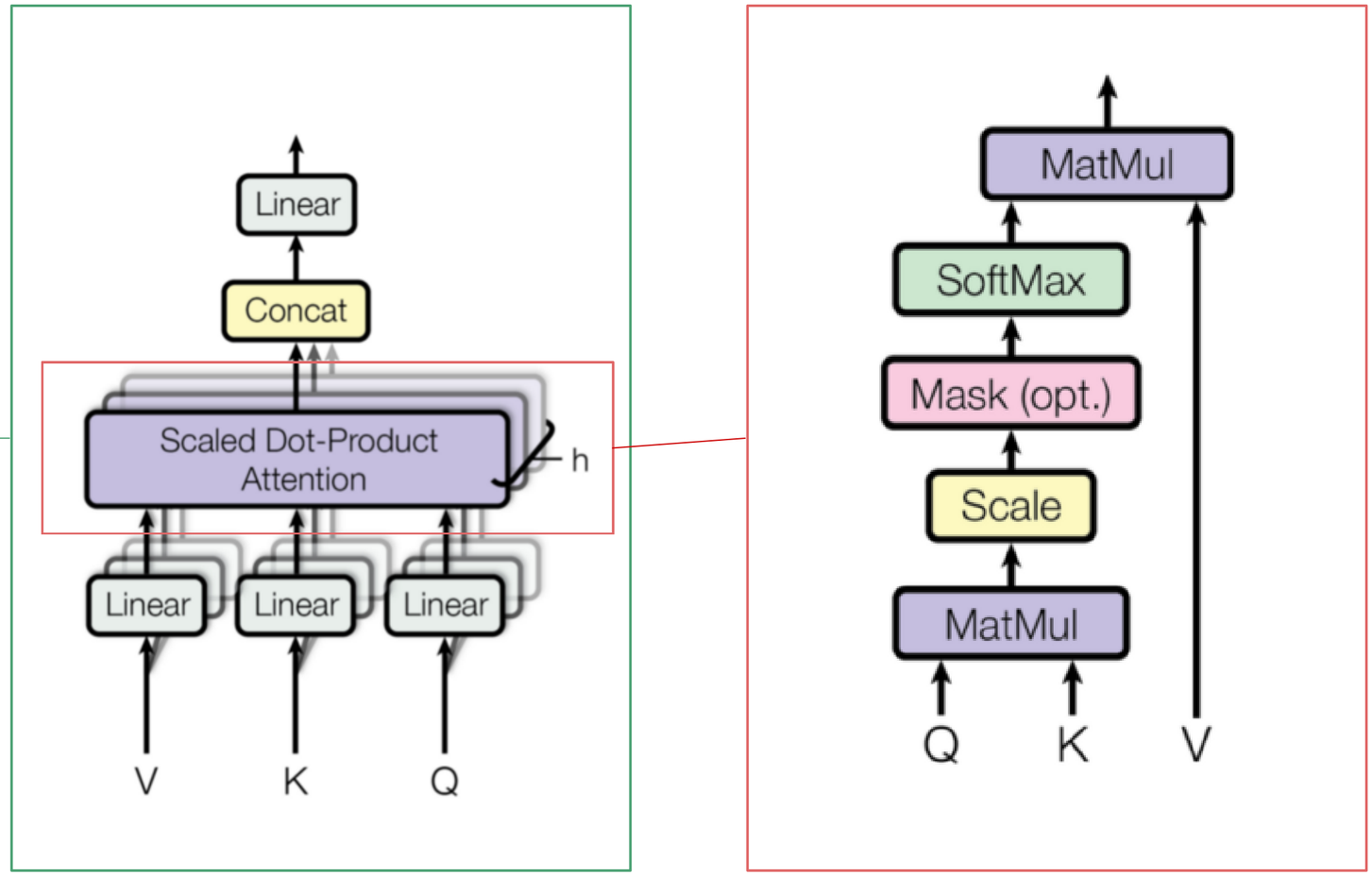
Encoder: Multi-head attention

Multi-head attention

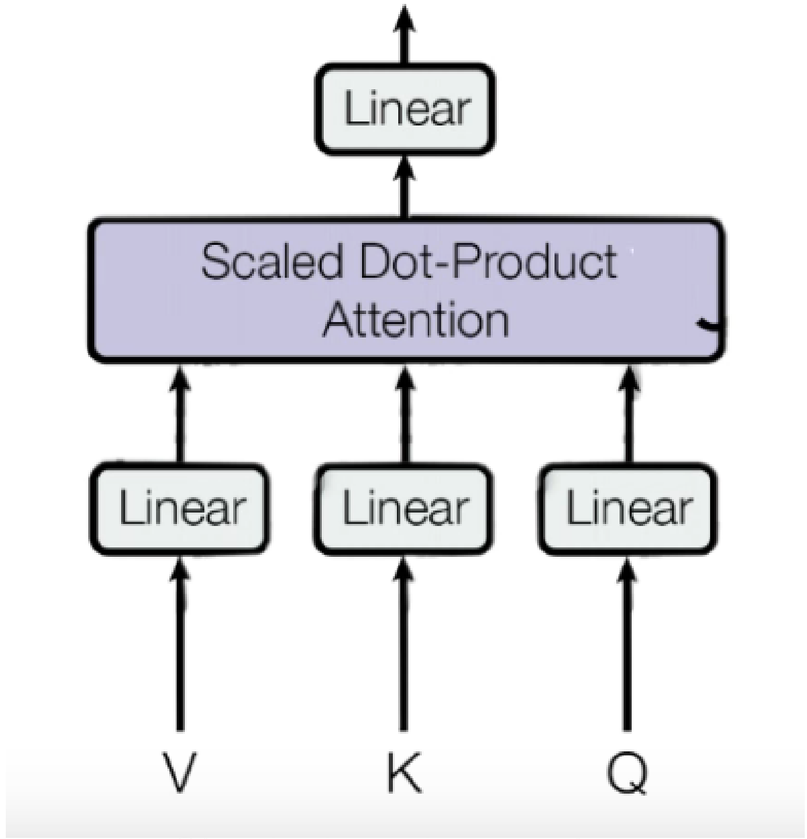


Encoder: Multi-head attention

Multi-head attention

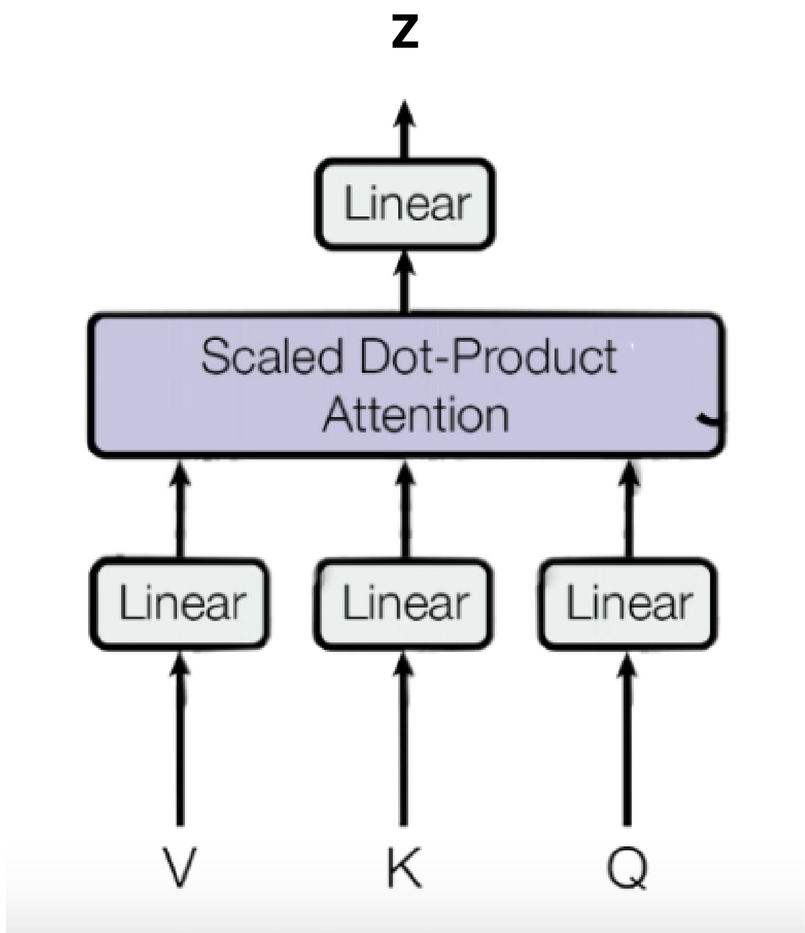


Encoder: Multi-head attention



Single-head attention

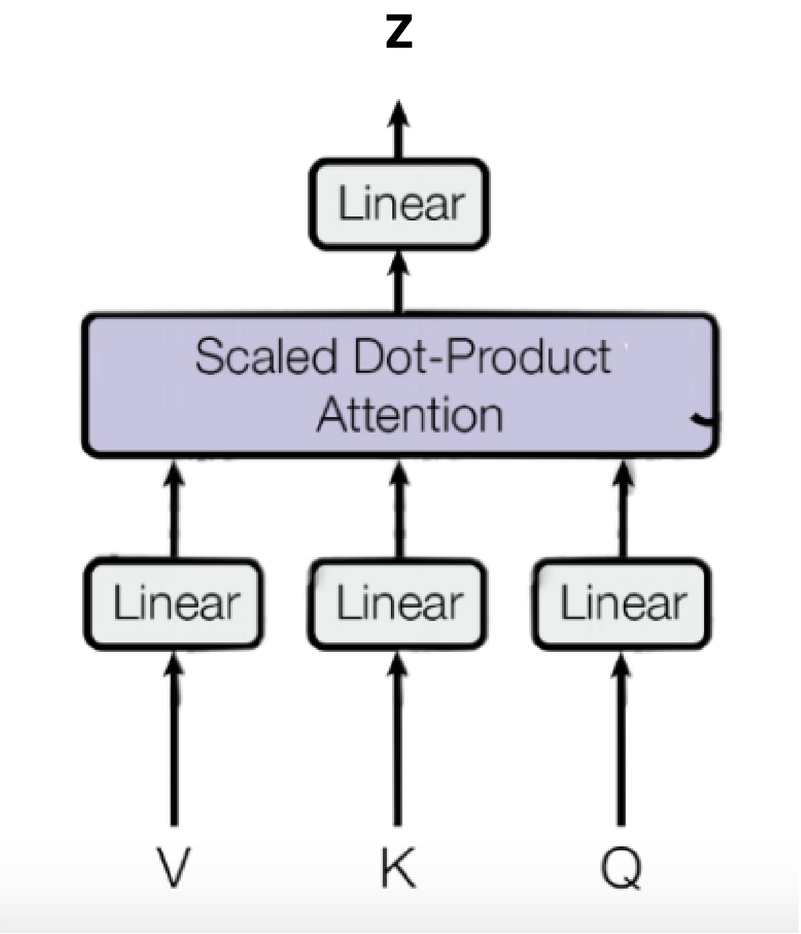
Encoder: Multi-head attention



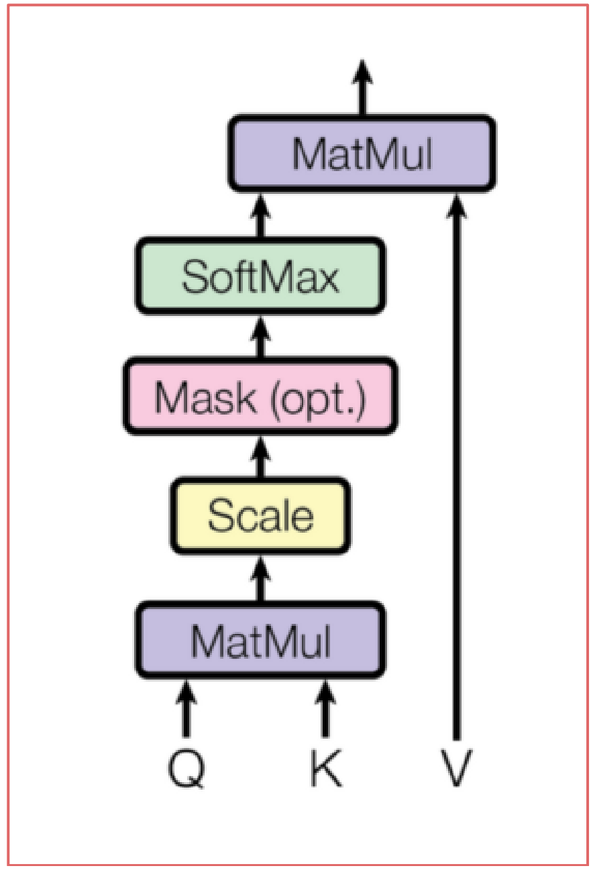
Single-head attention

$$Z = softmax\left(\frac{Q.K^T}{\sqrt{Dimension\ of\ vector\ Q, K\ or\ V}}\right).V$$

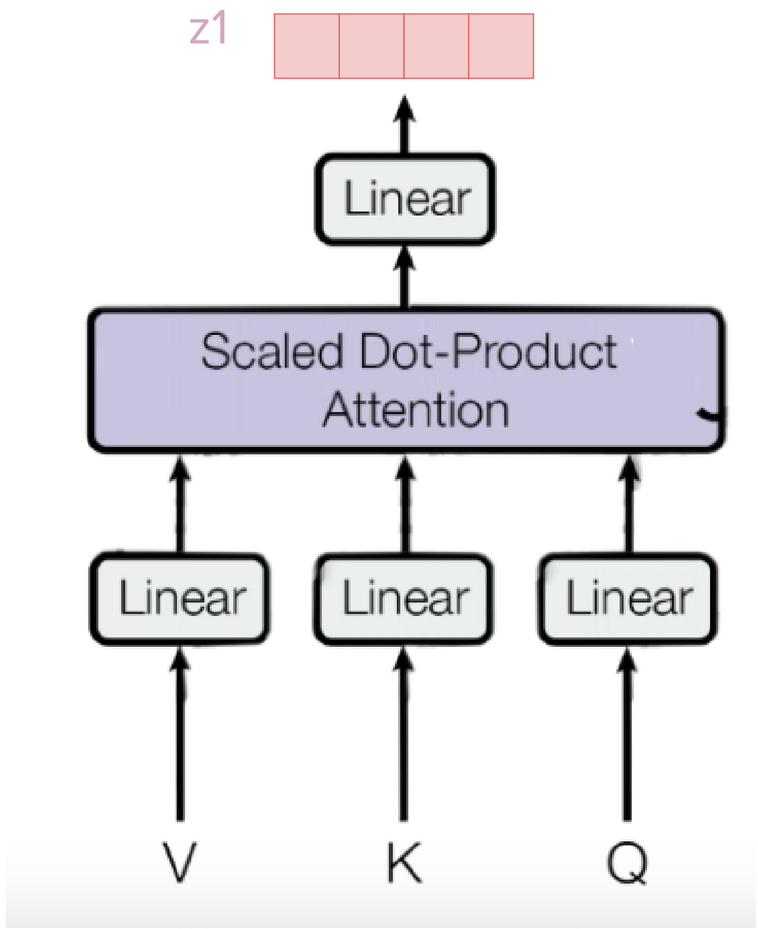
Encoder: Multi-head attention



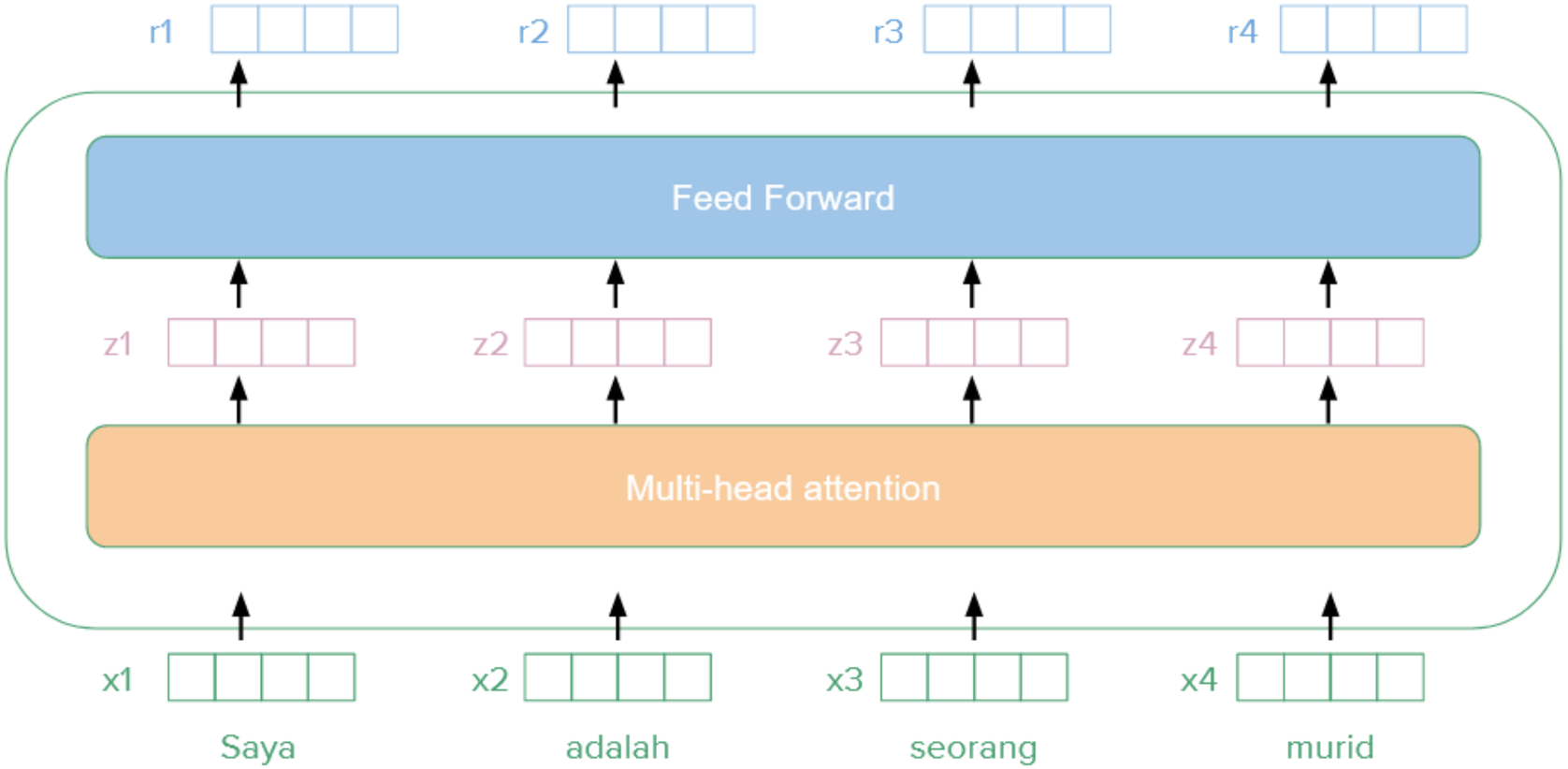
Single-head attention



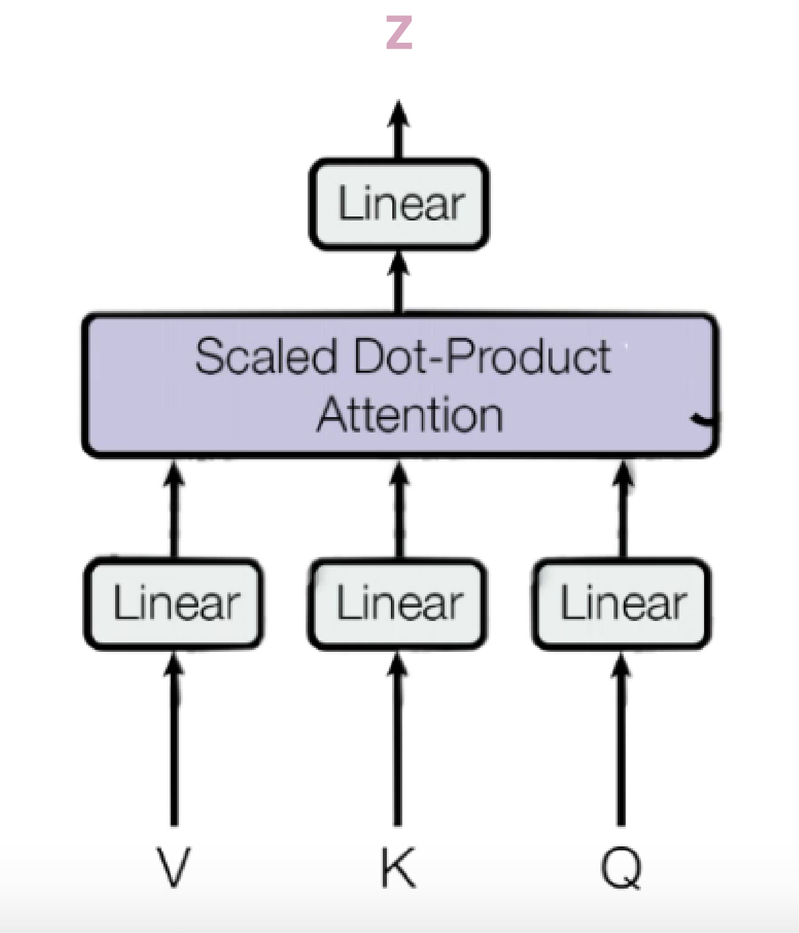
Encoder: Multi-head attention



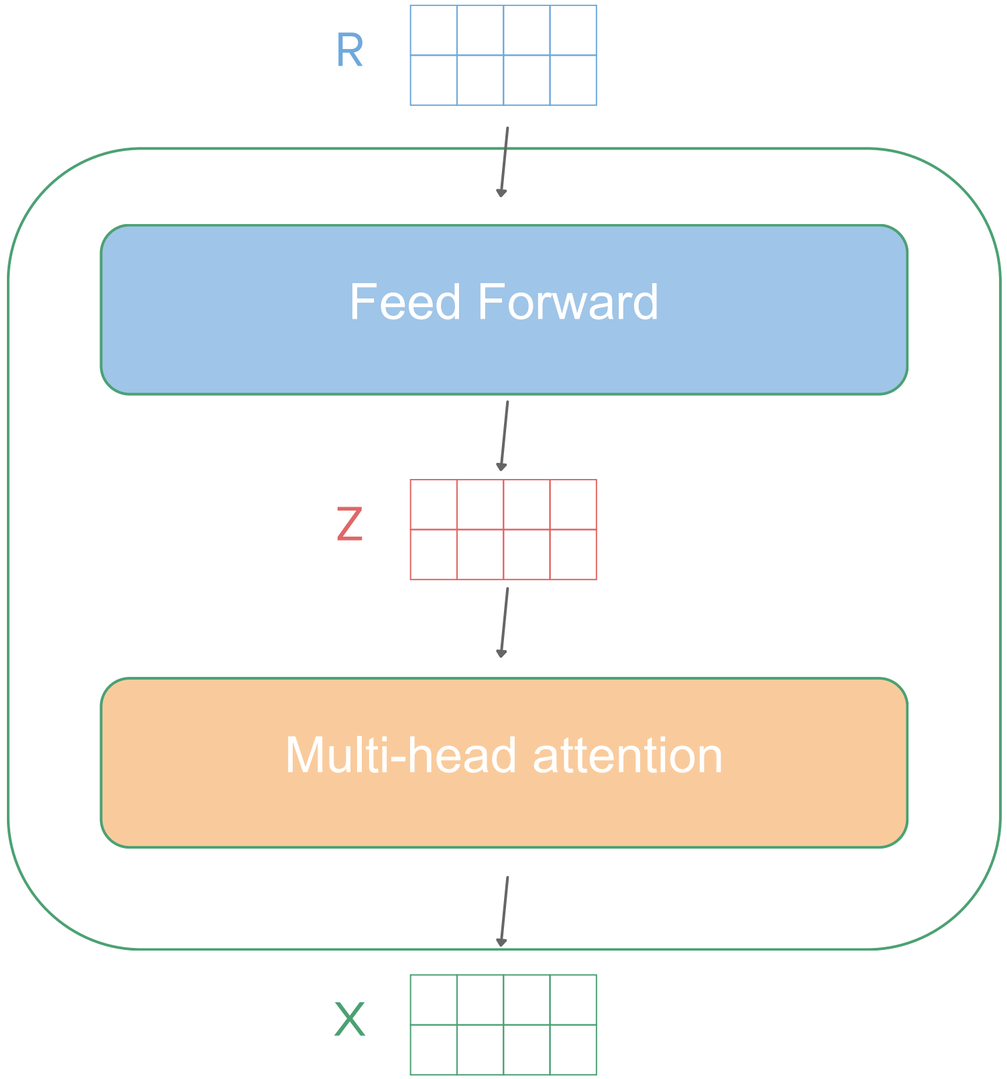
Single-head attention



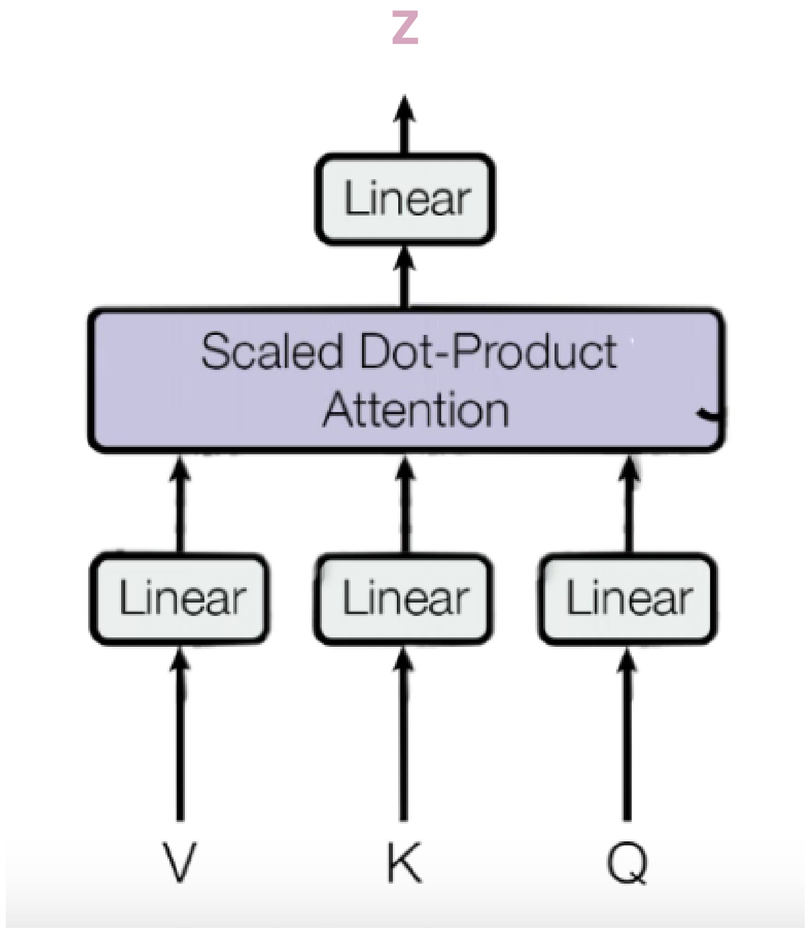
Encoder: Multi-head attention



Single-head attention

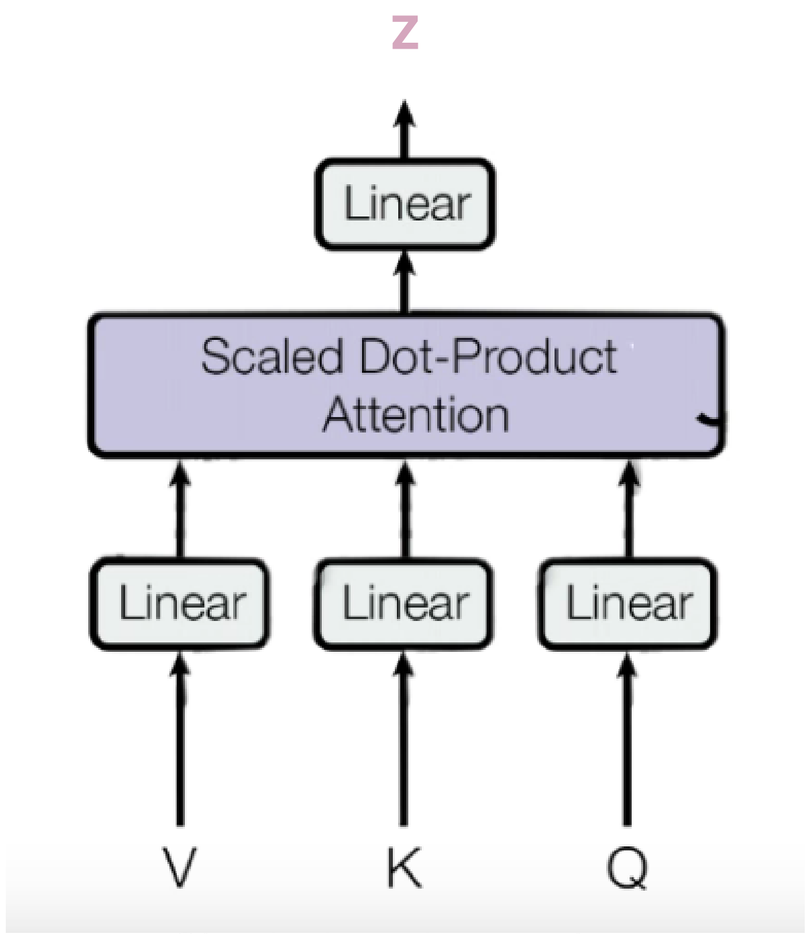


Encoder: Multi-head attention



Single-head attention

Encoder: Multi-head attention



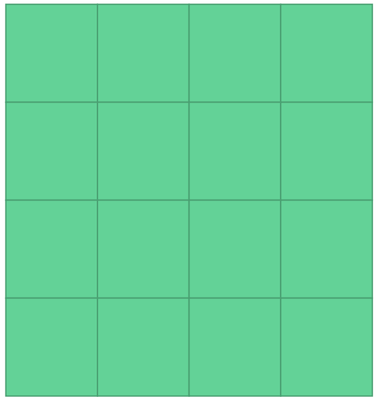
Single-head attention

Saya

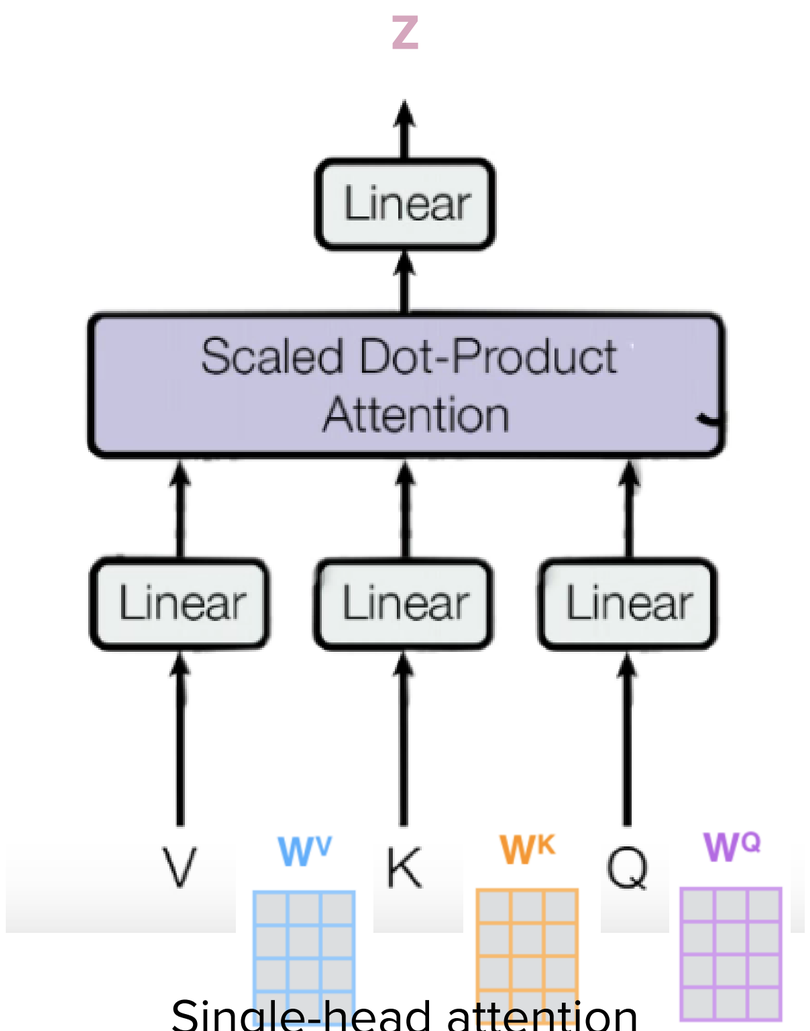
adalah

seorang

murid



Encoder: Multi-head attention

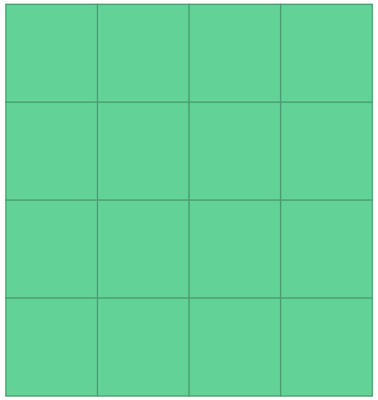


Saya

adalah

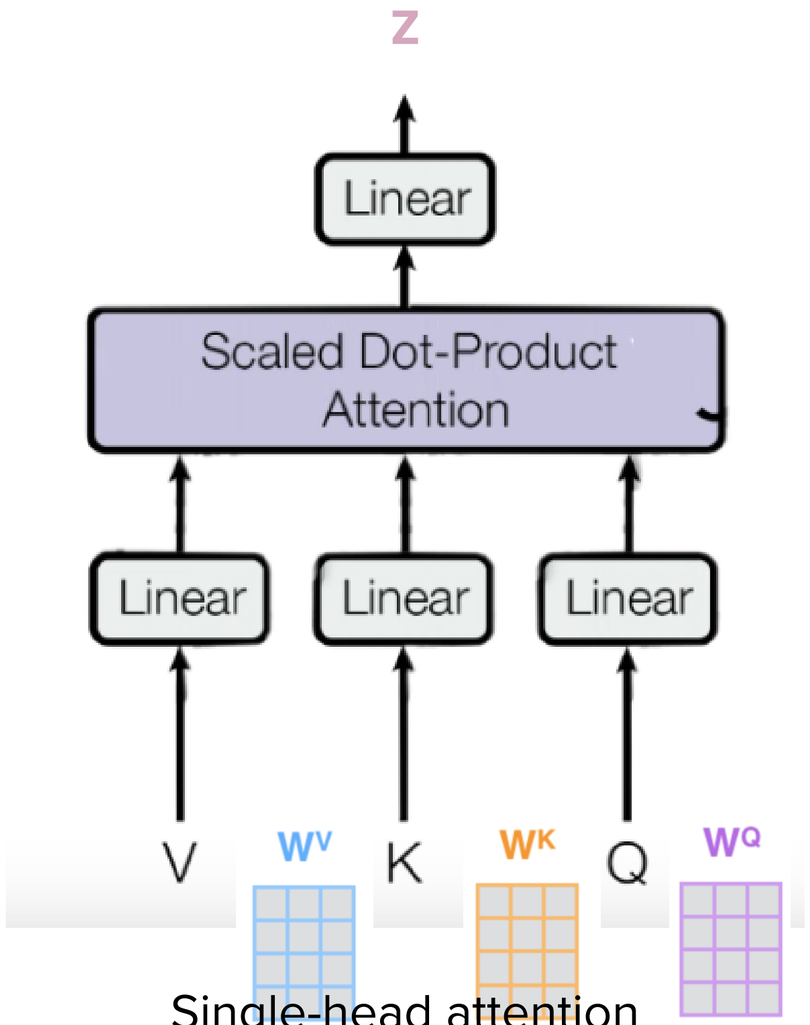
seorang

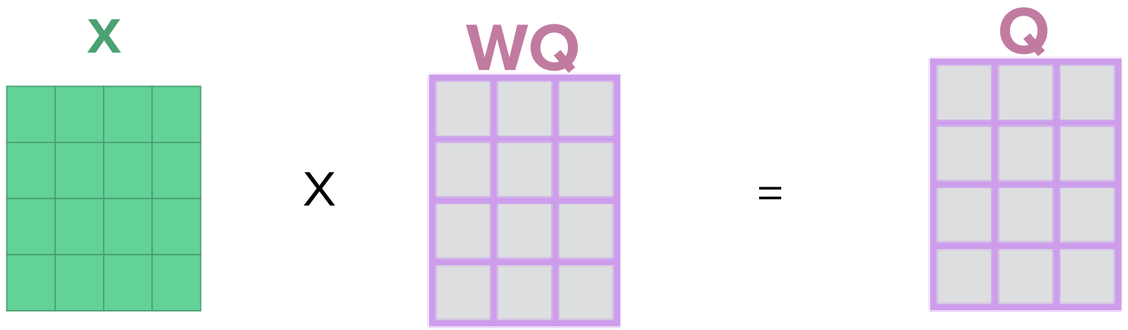
murid



Single-head attention

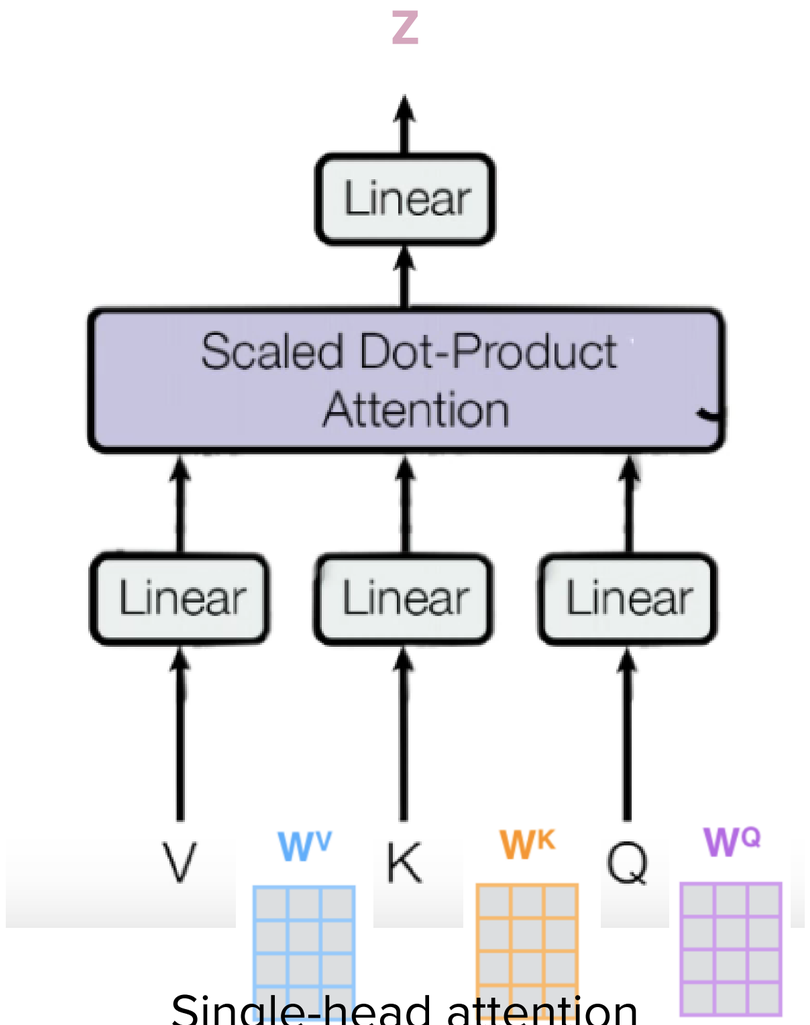
Encoder: Multi-head attention



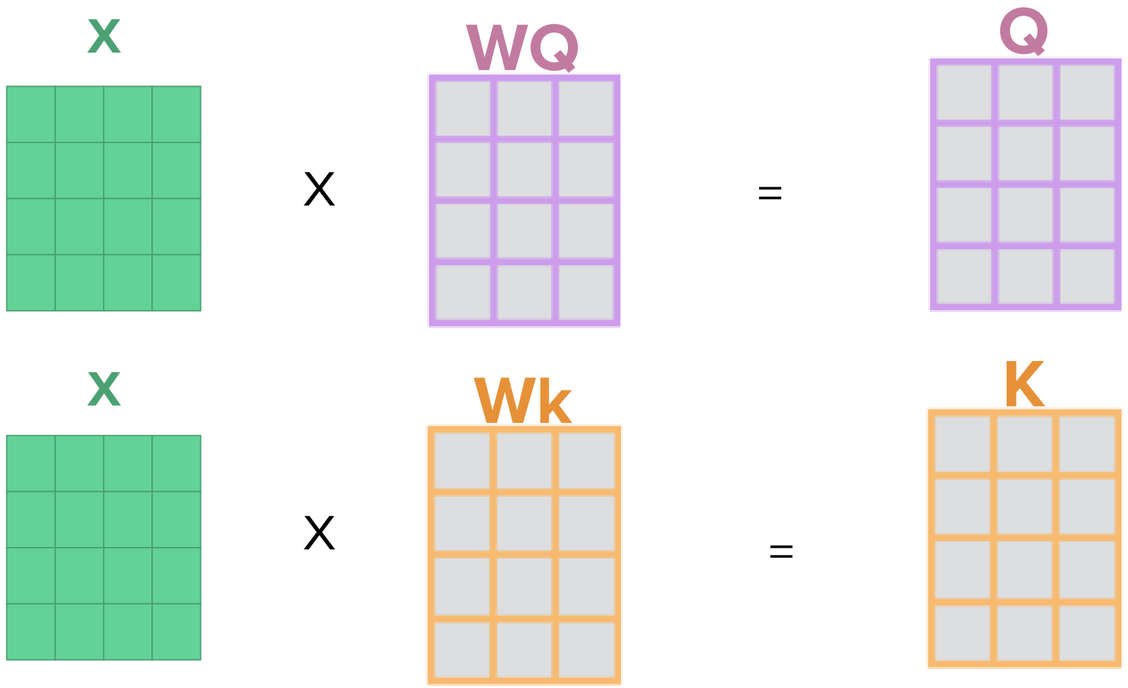


Single-head attention

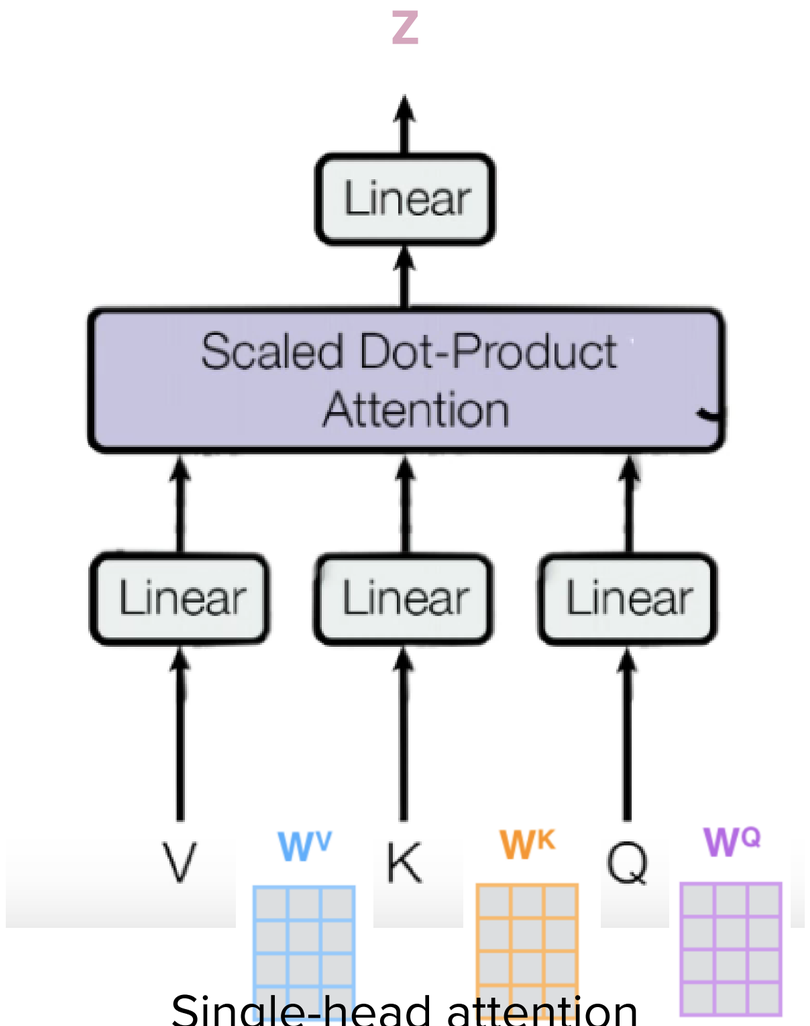
Encoder: Multi-head attention



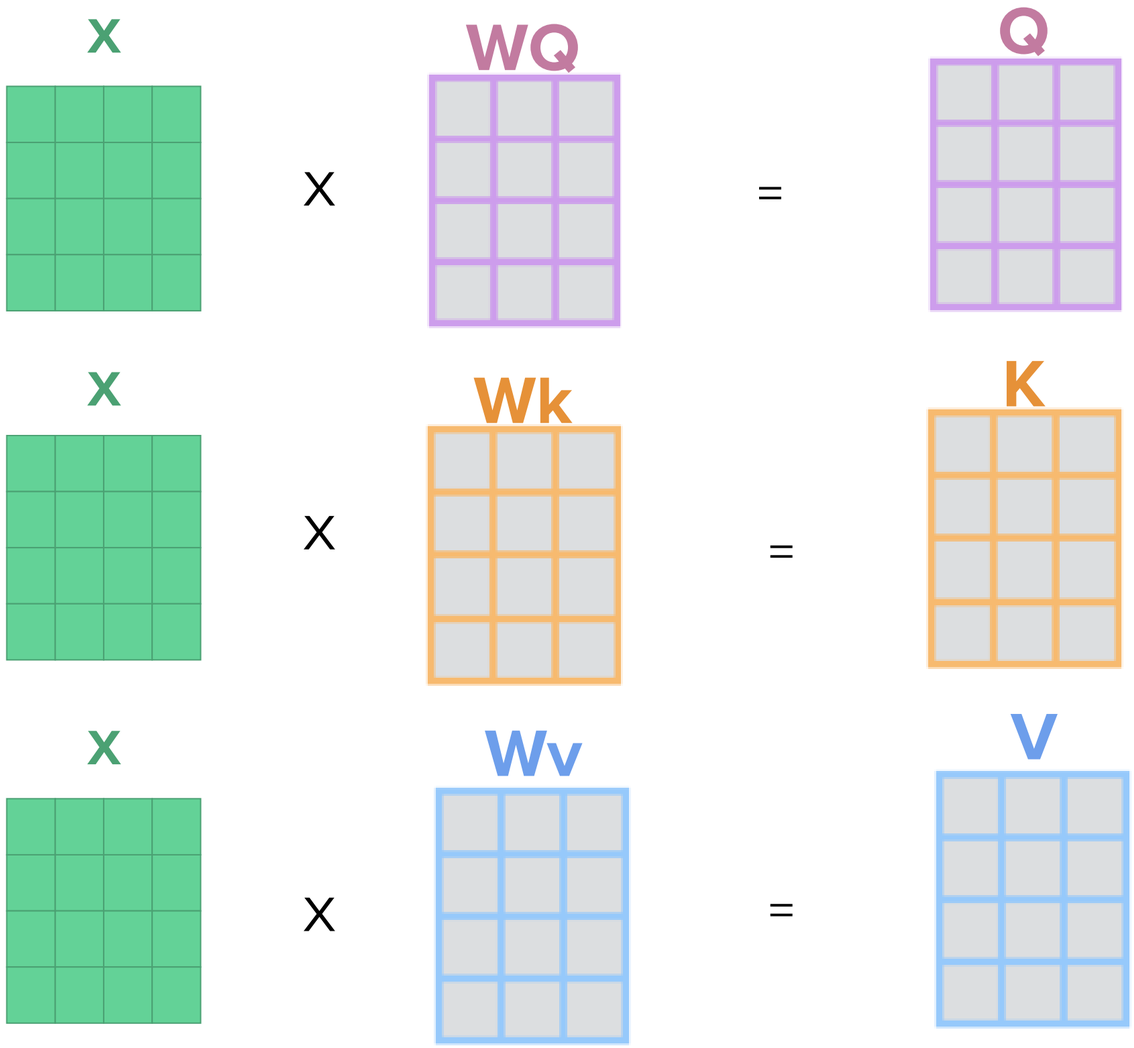
Single-head attention



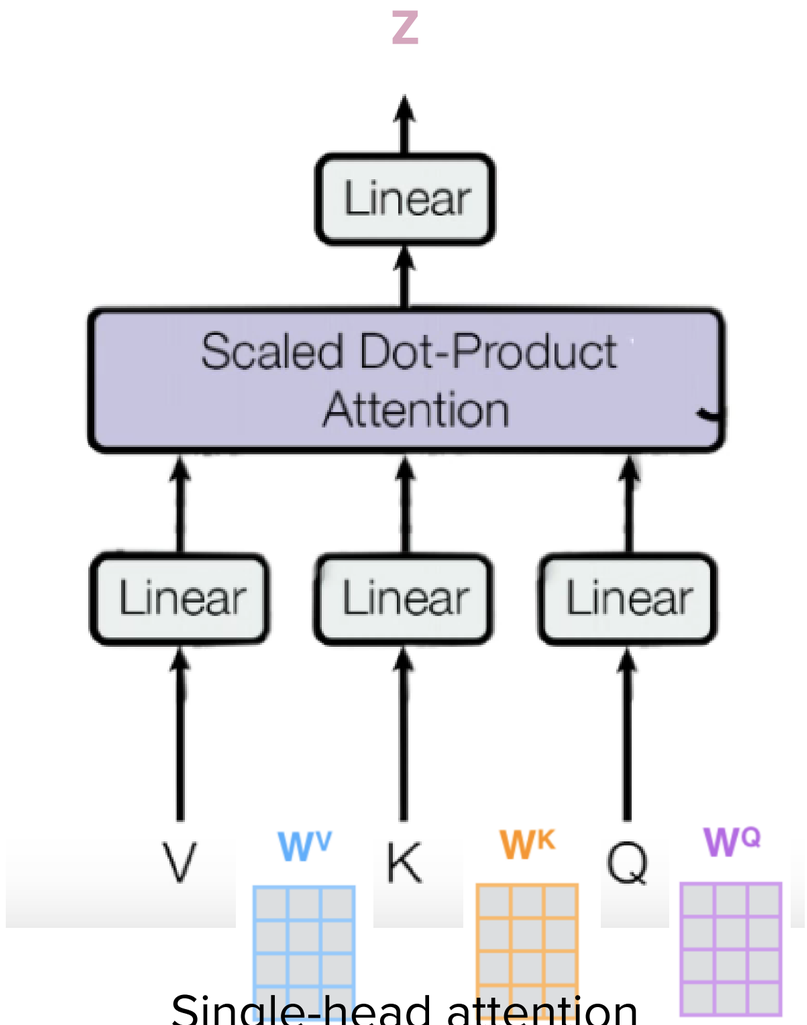
Encoder: Multi-head attention



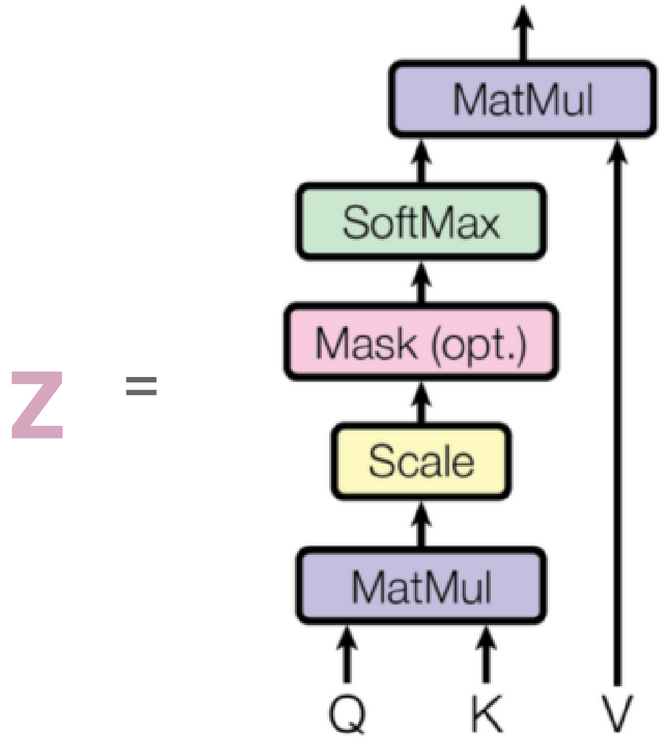
Single-head attention



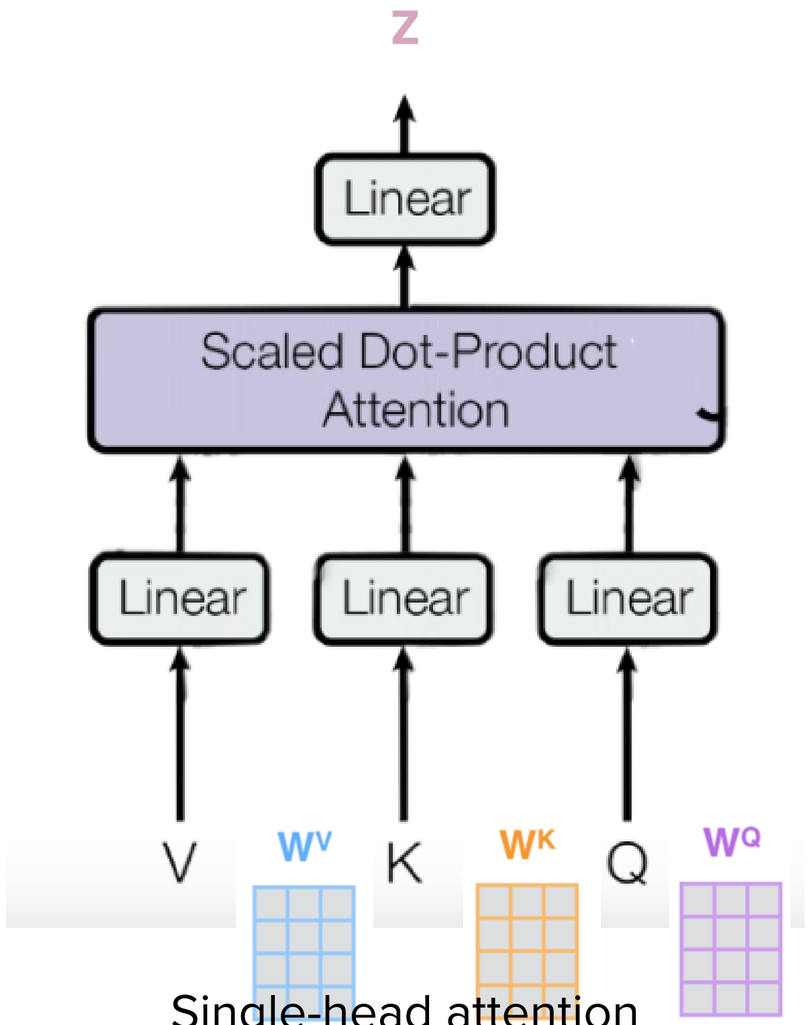
Encoder: Multi-head attention



Single-head attention



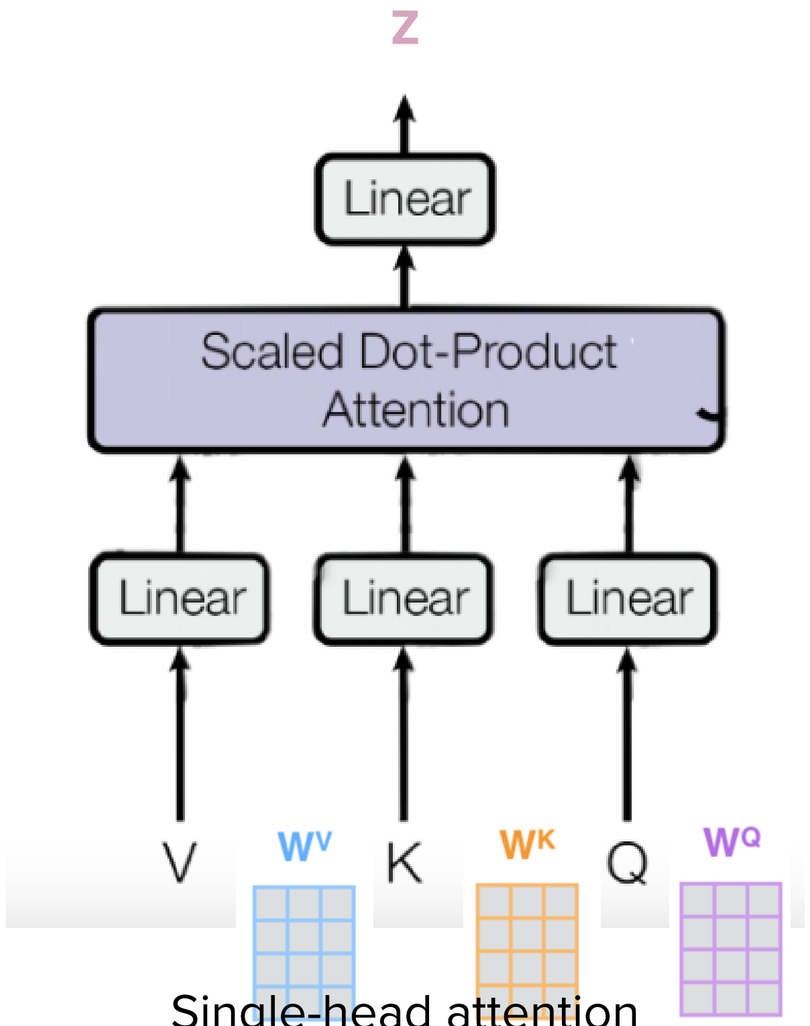
Encoder: Multi-head attention



$$Z = softmax\left(\frac{Q.K^T}{\sqrt{Dimension\ of\ vector\ Q, K\ or\ V}}\right).V$$

Single-head attention

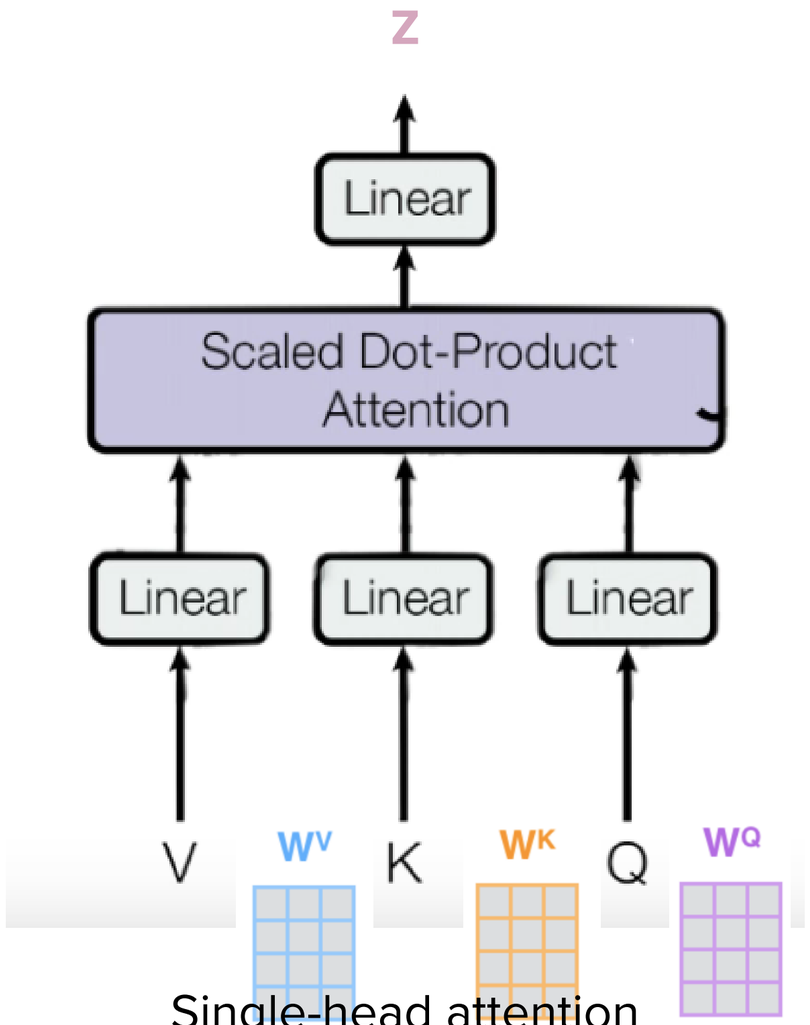
Encoder: Multi-head attention



$$Z = \text{Softmax}\left(\frac{Q K^T}{\sqrt{d}}\right) V$$

Single-head attention

Encoder: Multi-head attention

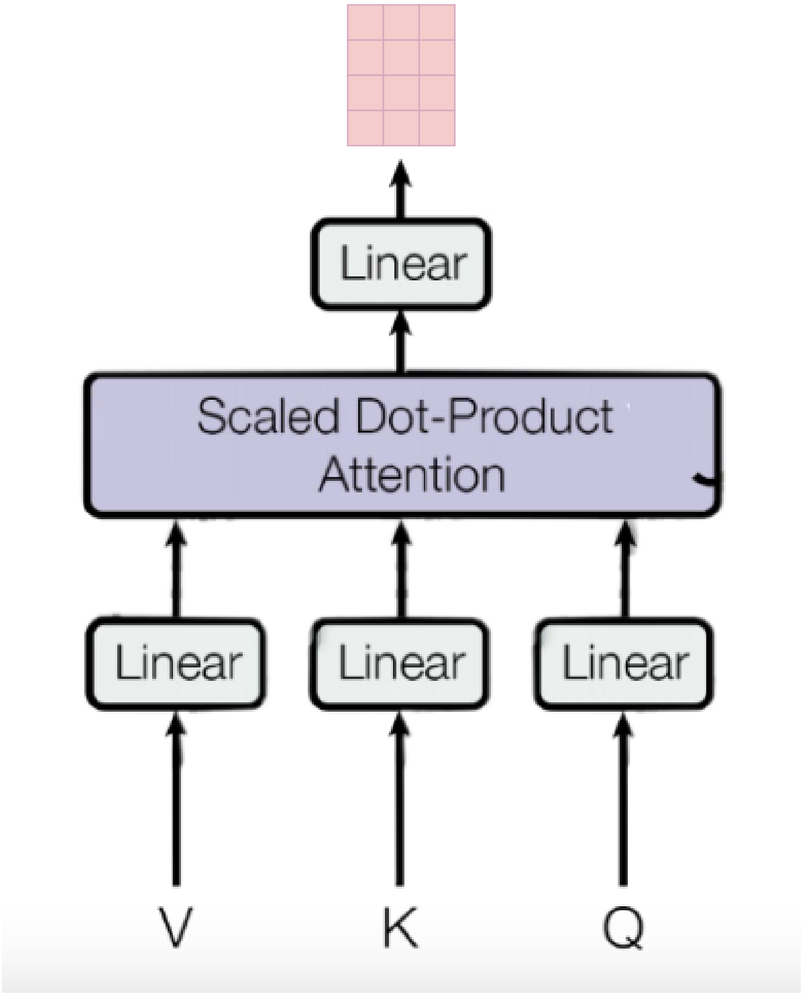


$$Z = \text{Softmax}\left(\frac{Q K^T}{\sqrt{d}}\right) V$$

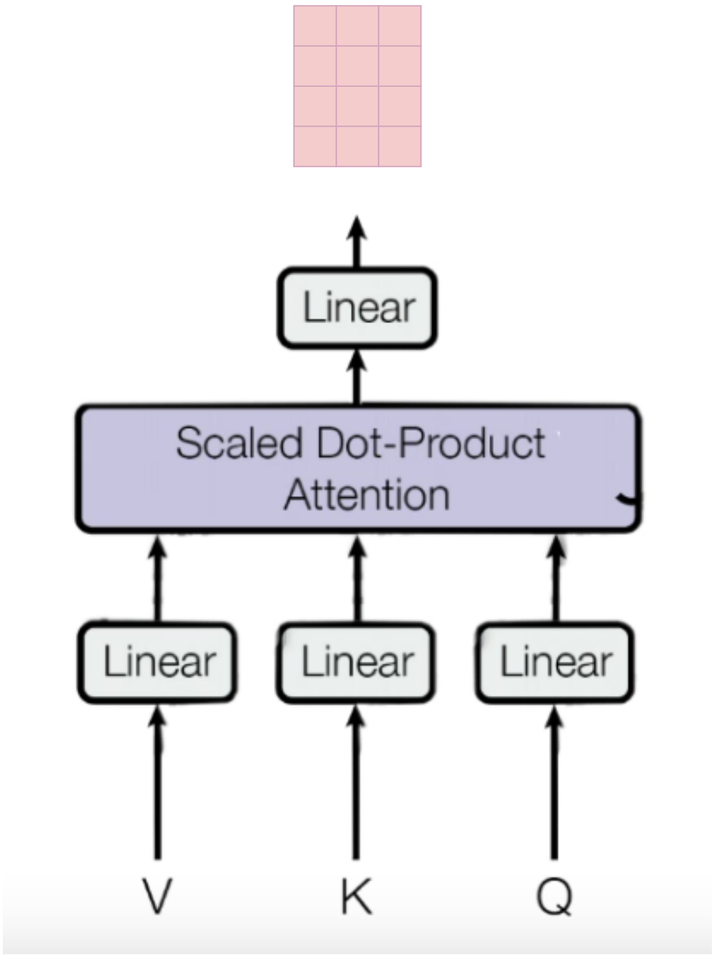
$$Z = \begin{bmatrix} & & \\ & & \\ & & \\ & & \end{bmatrix}$$

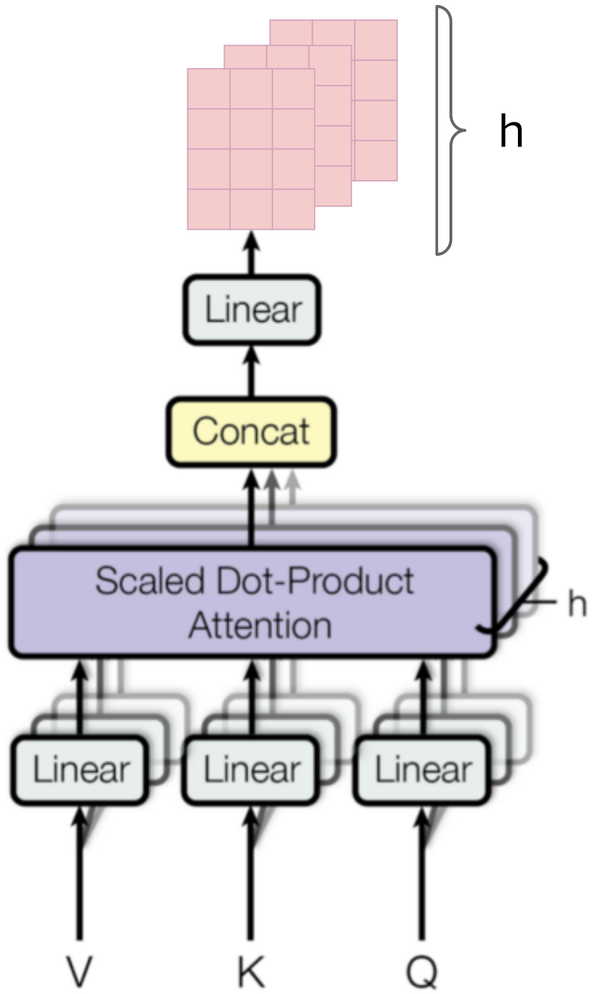
Single-head attention

Encoder: Multi-head attention

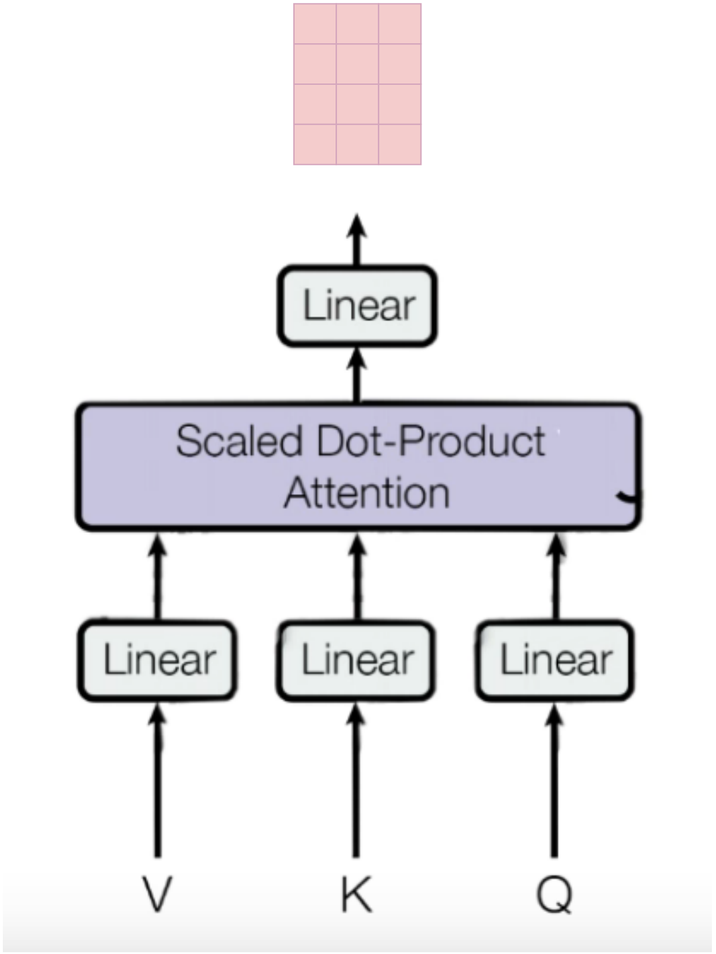


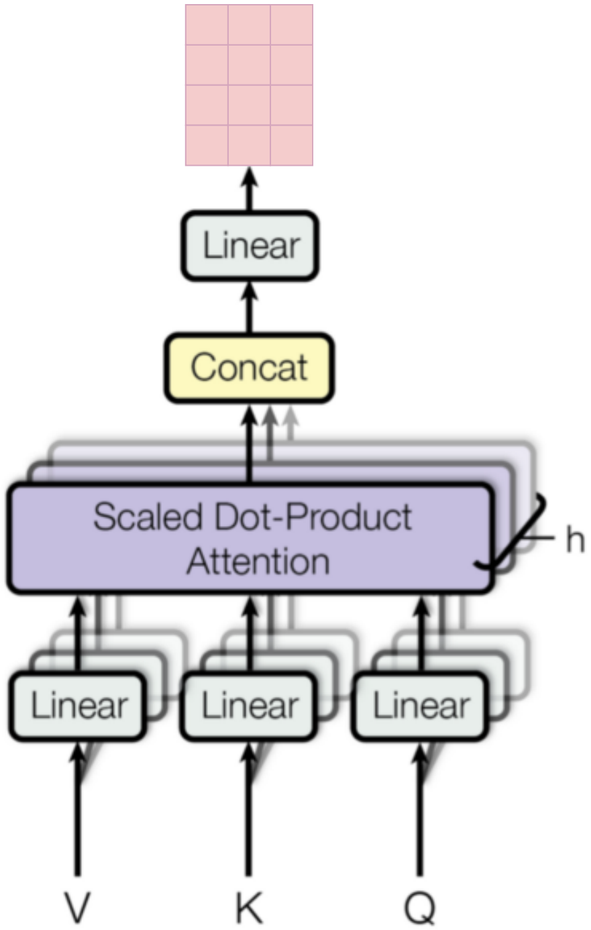
Encoder: Multi-head attention





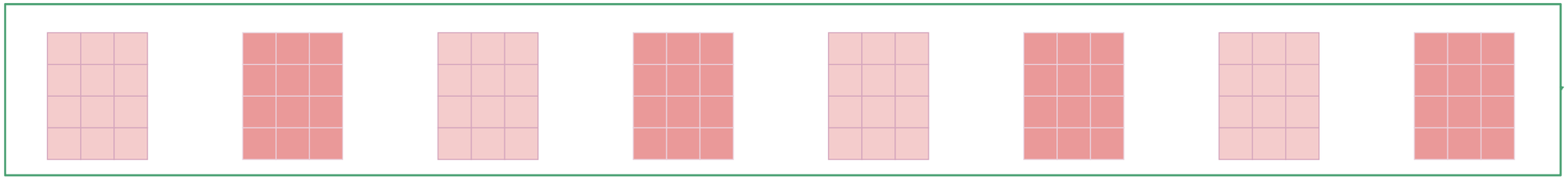
Encoder: Multi-head attention

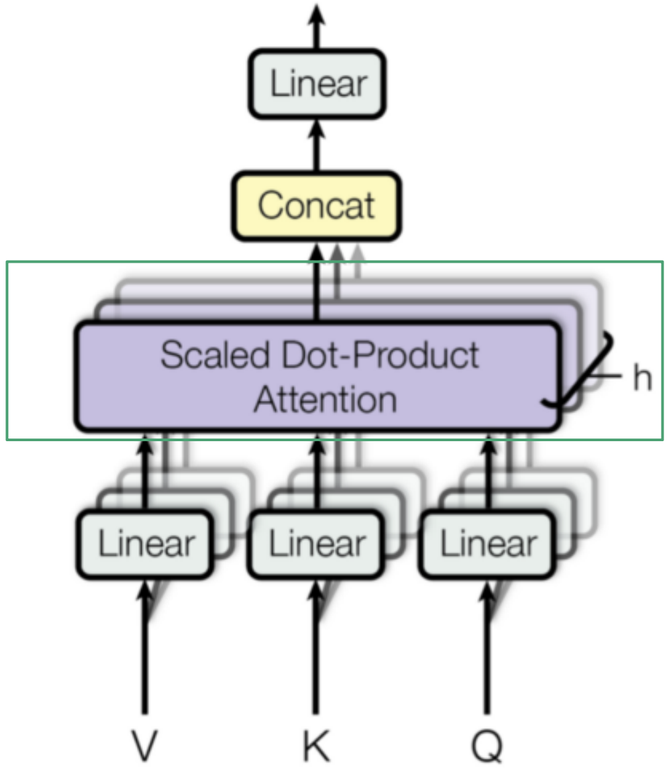




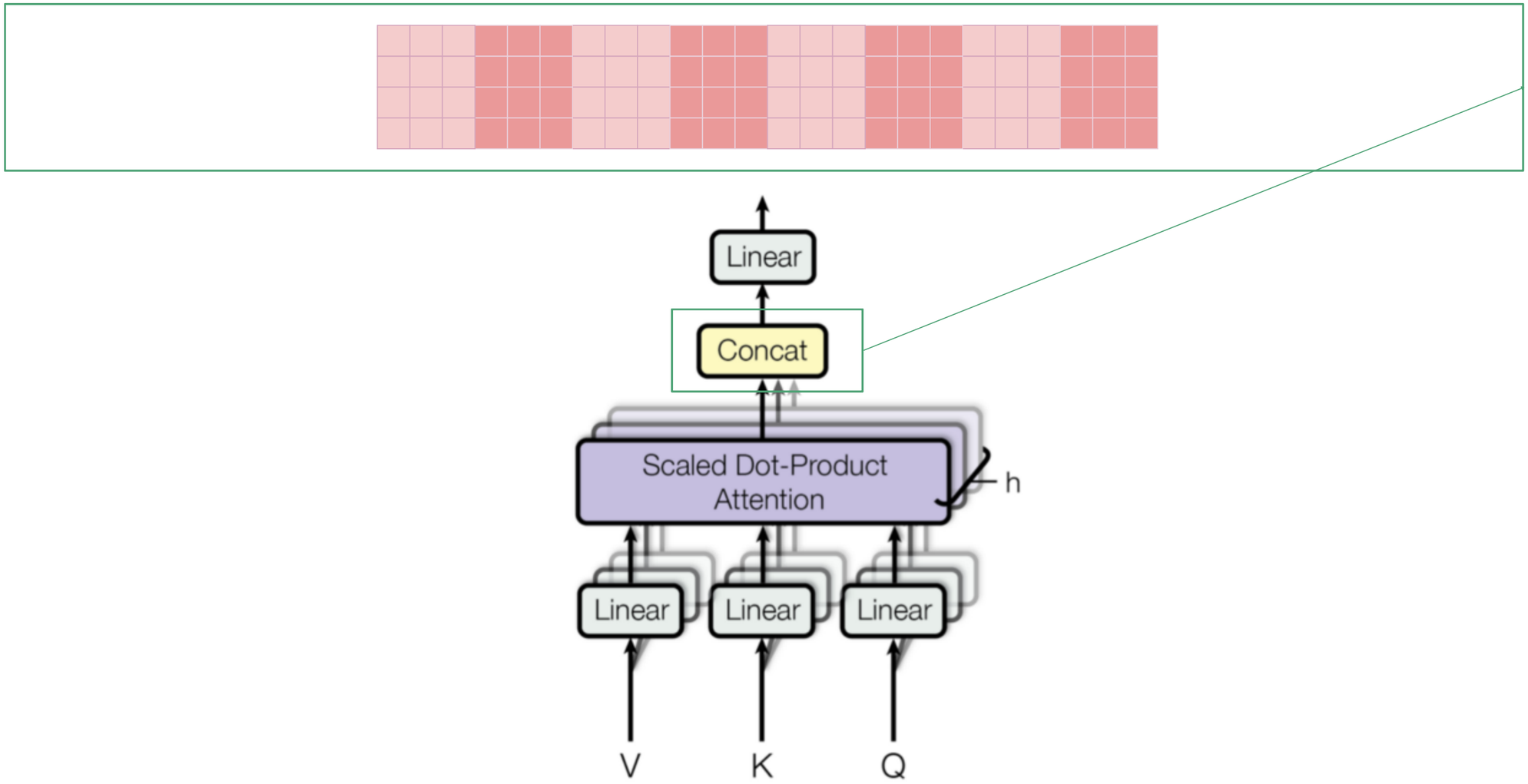
h = 8

Encoder: Multi-head attention

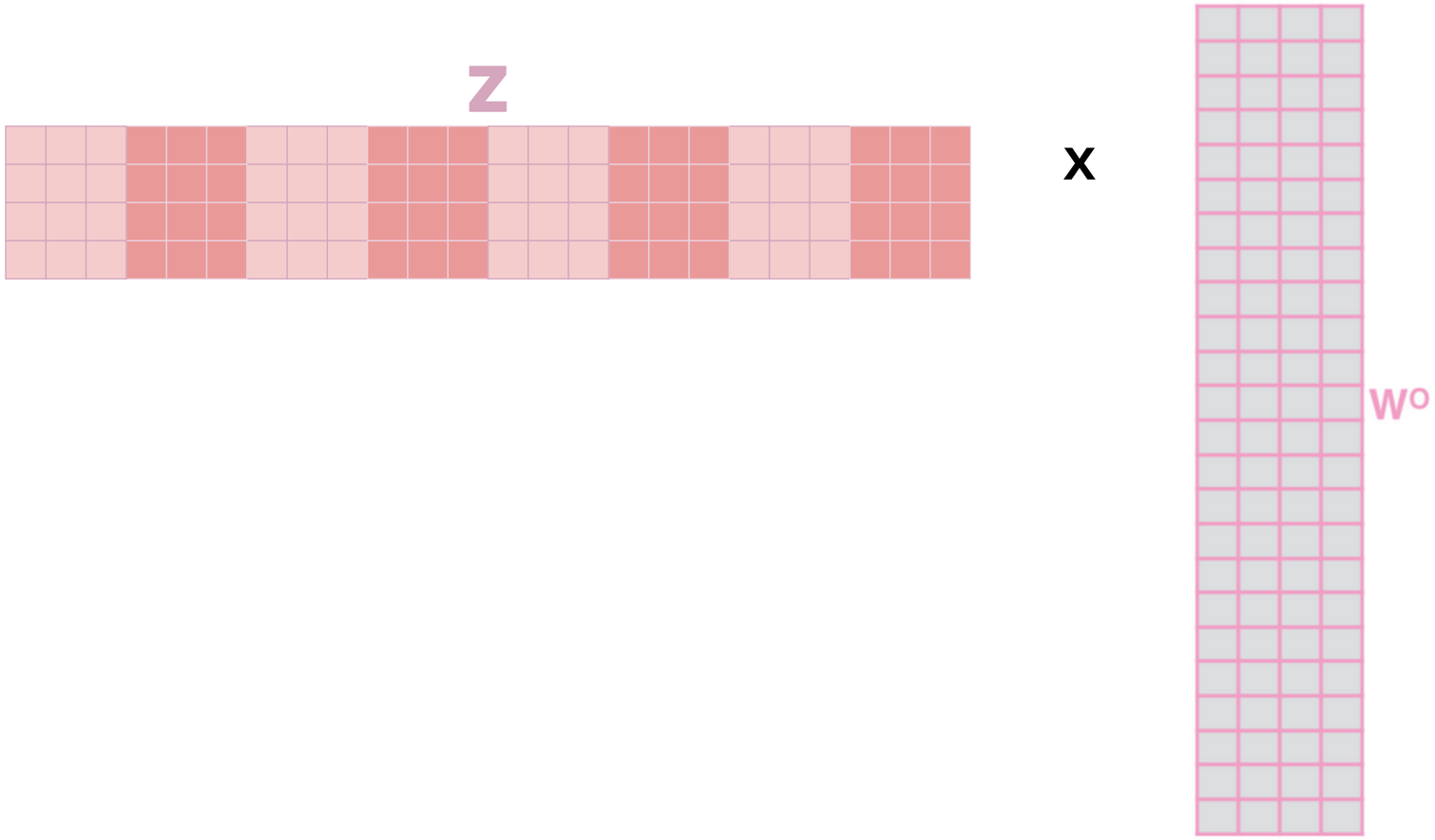




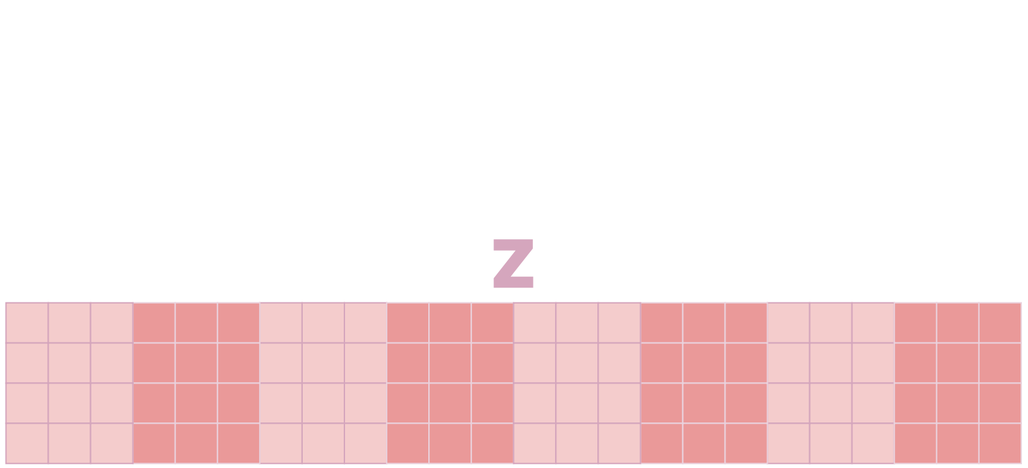
Encoder: Multi-head attention



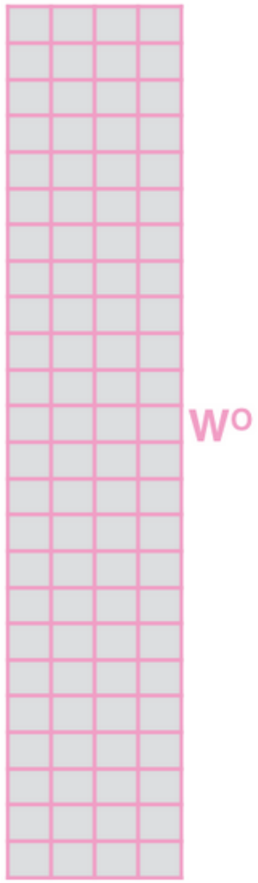
Encoder: Multi-head attention



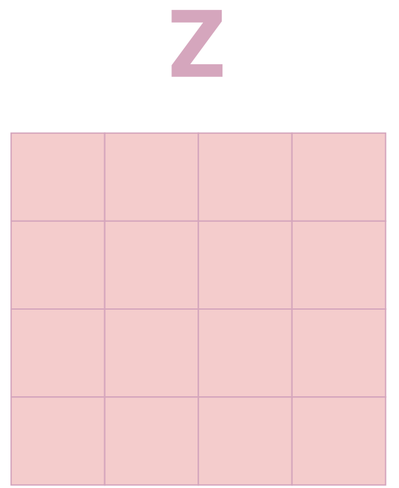
Encoder: Multi-head attention

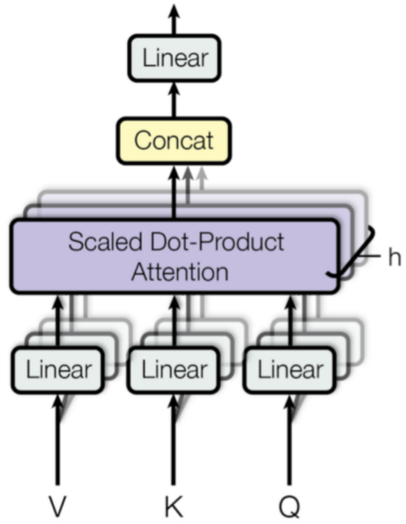


x



=





Encoder: Multi-head attention (summary)

1) This is our input sentence*

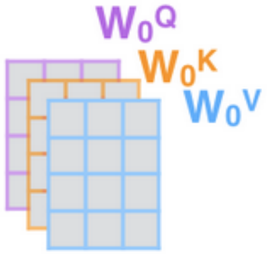
2) We embed each word*

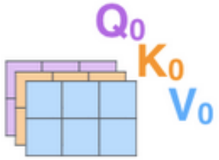
3) Split into 8 heads. We multiply X or R with weight matrices

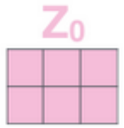
4) Calculate attention using the resulting Q/K/V matrices

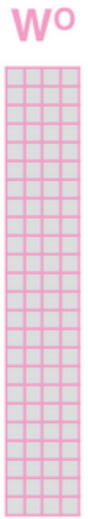
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

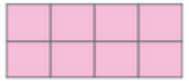
Thinking Machines



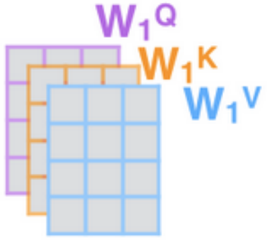


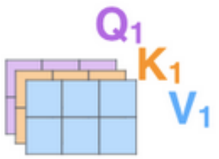


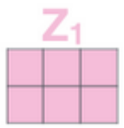


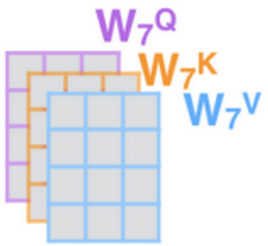


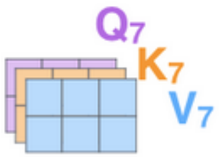
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

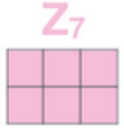




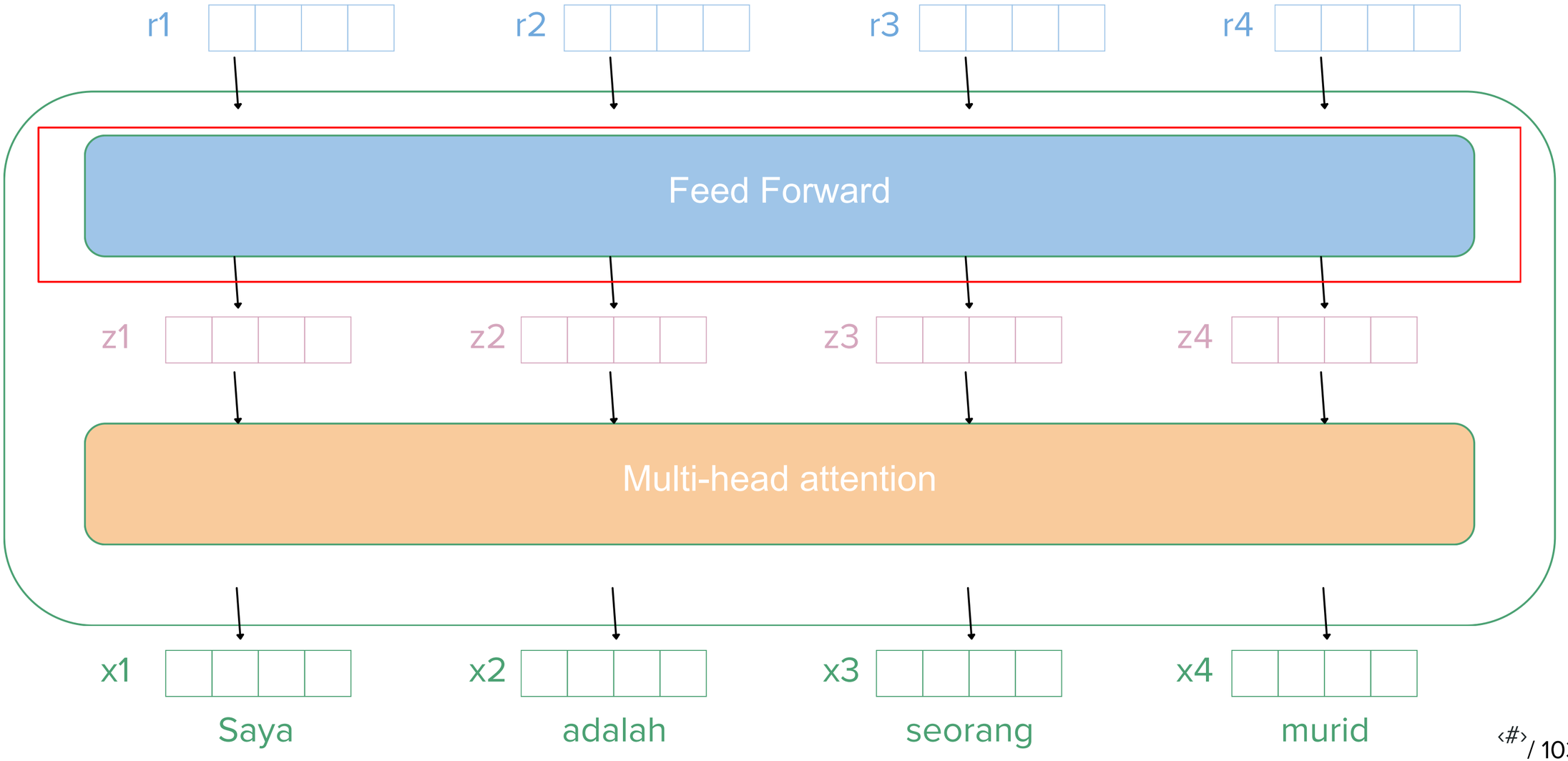








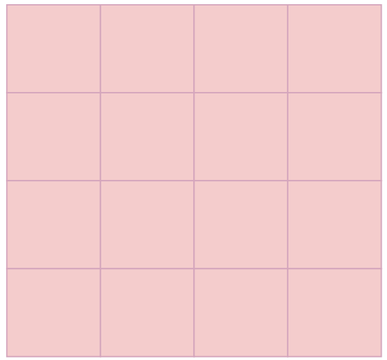
Encoder: Feed Forward

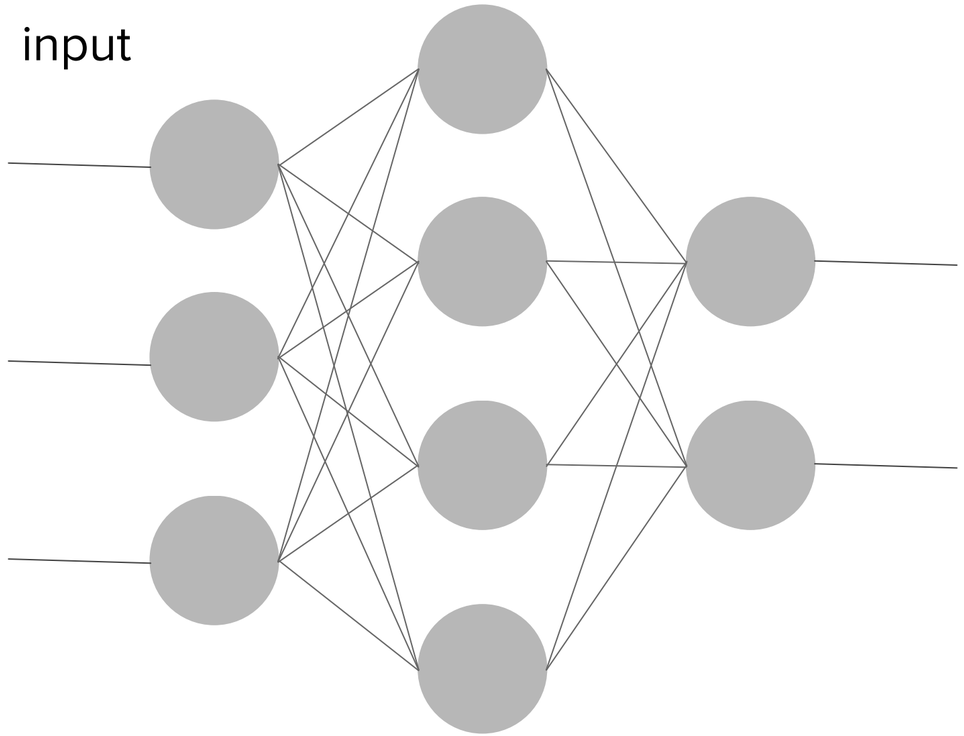


Encoder: Feed Forward

Feed Forward

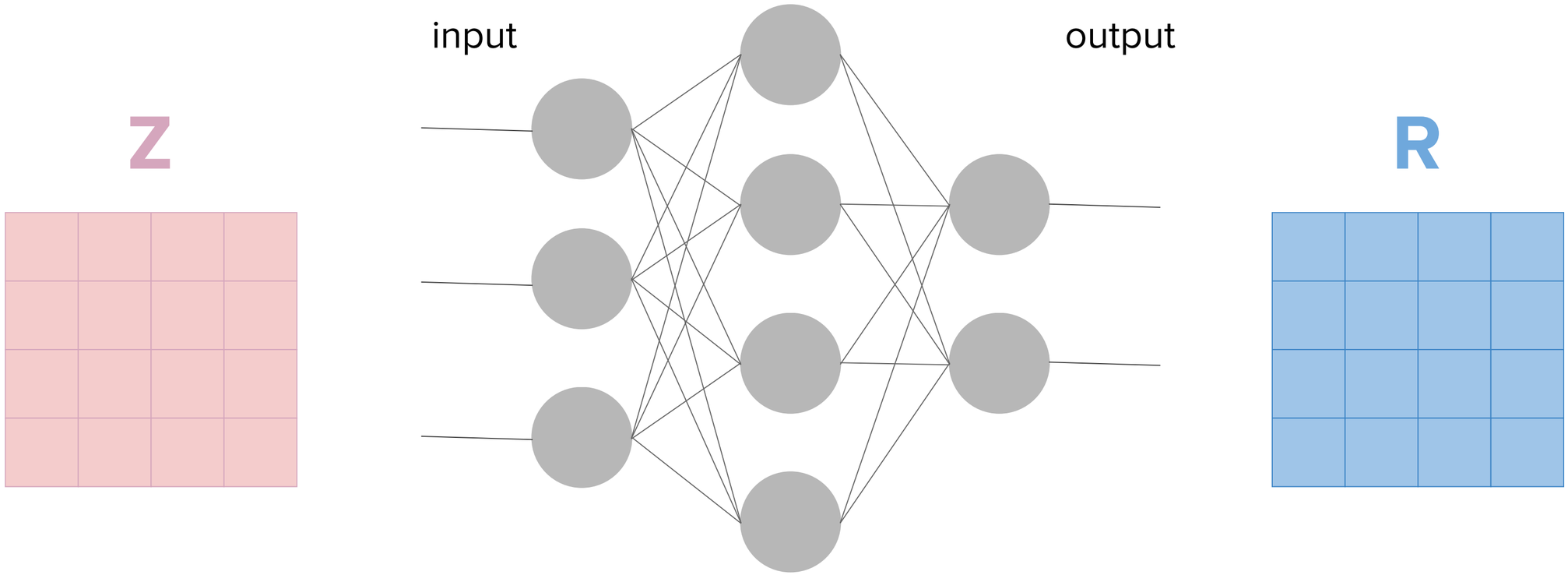
Z





Encoder: Feed Forward

Feed Forward

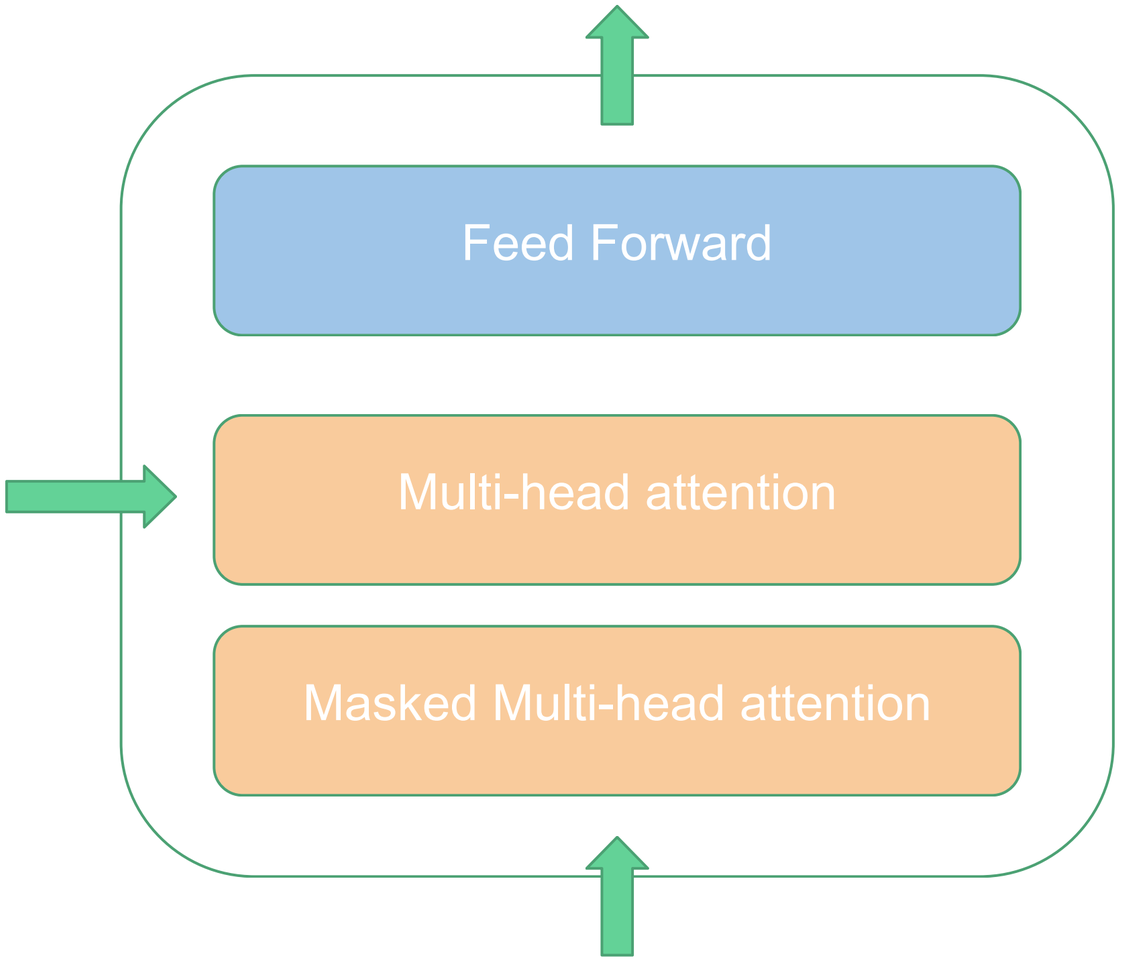


Any Question Guys ~

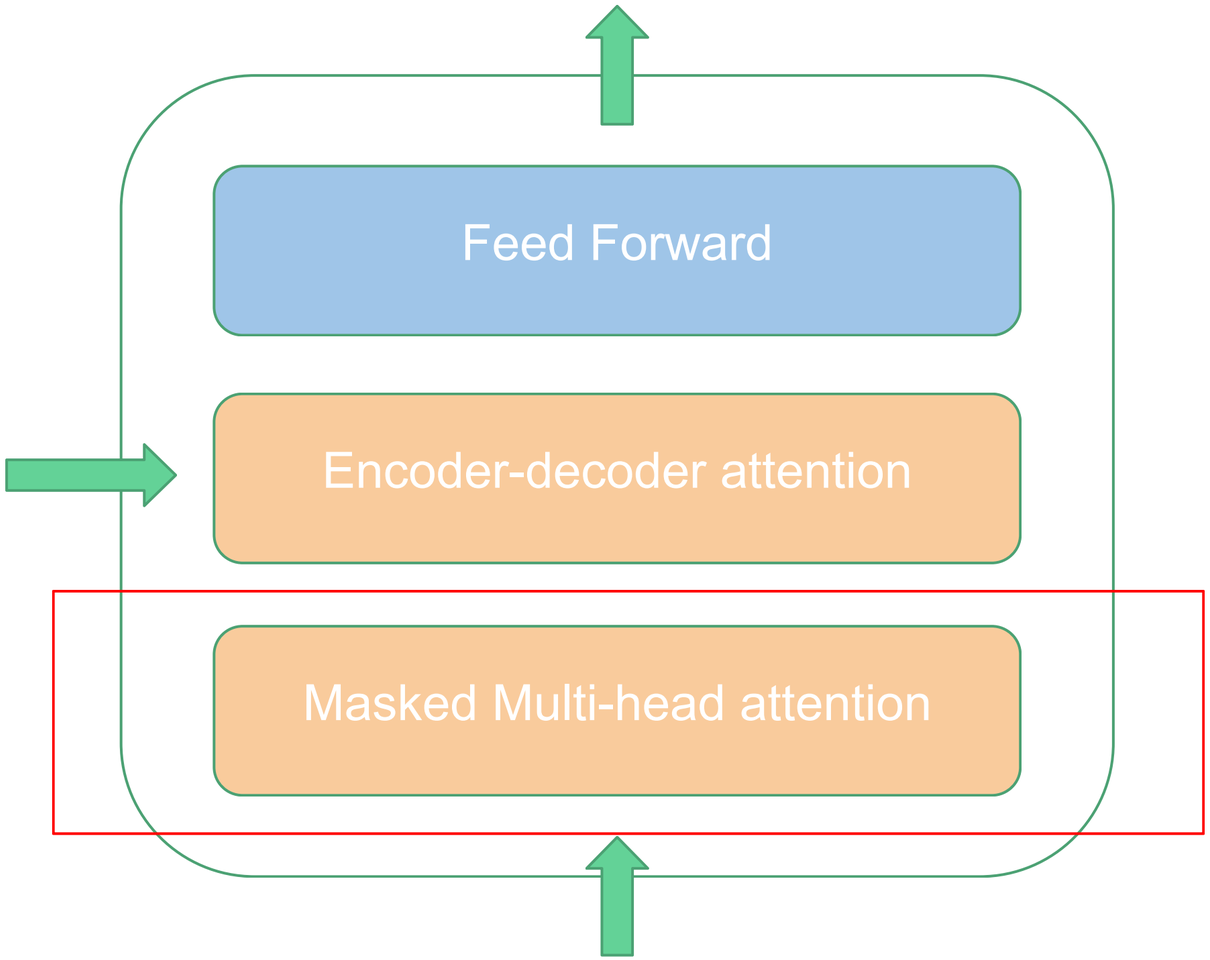
Transformer's Decoder

Indonesia AI

Decoder


Feed Forward
Multi-head attention
Masked Multi-head attention

Decoder

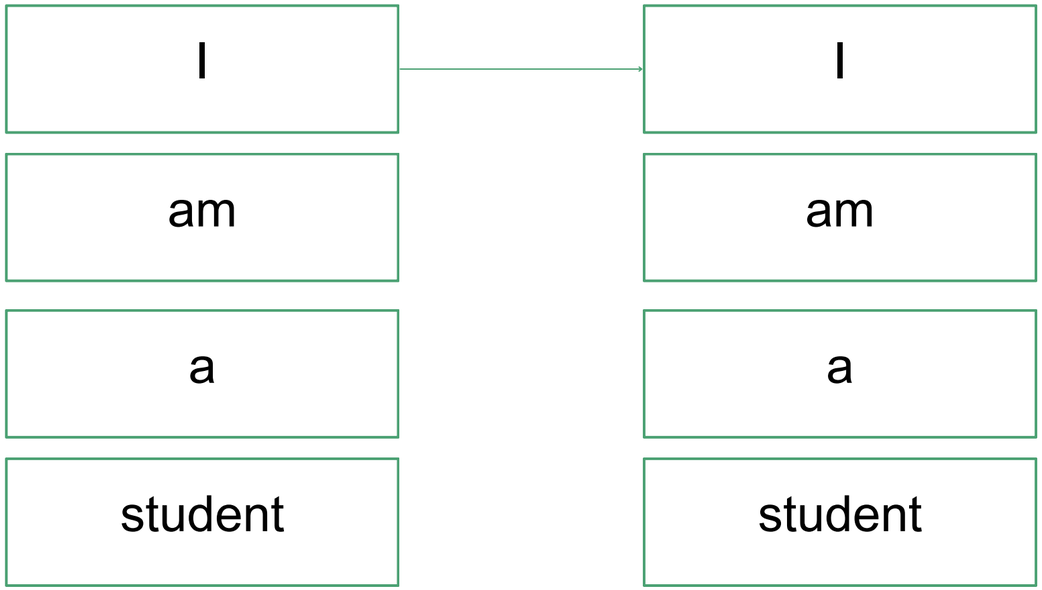

Feed Forward
Encoder-decoder attention
Masked Multi-head attention

Decoder: Masked Multi-head attention

Very similar with multi-head attention in **encoder**

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words and another previous words in a sentence

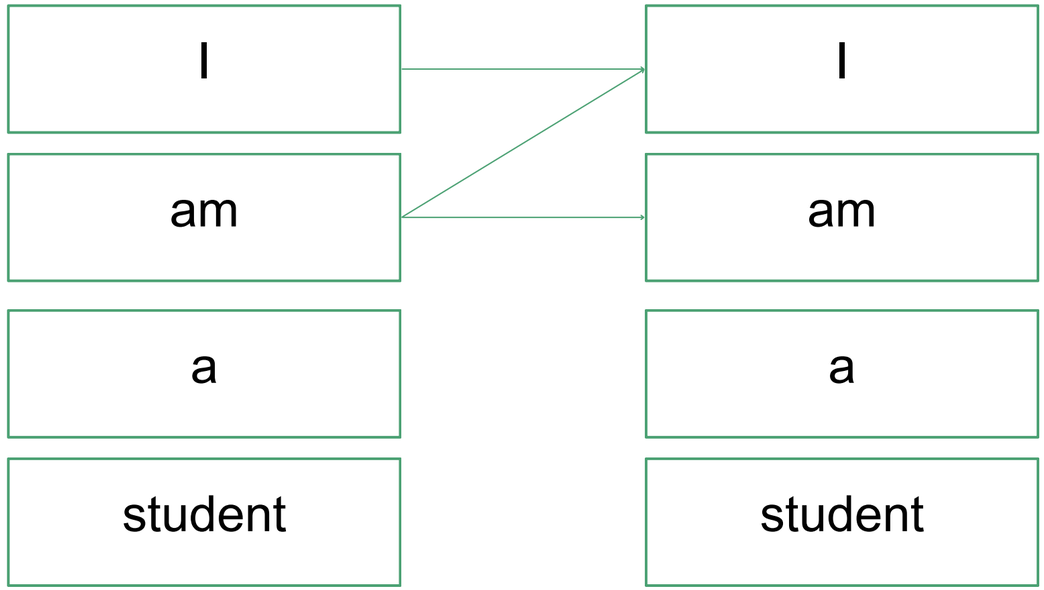


Decoder: Masked Multi-head attention

Very similar with multi-head attention in **encoder**

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words and another previous words in a sentence

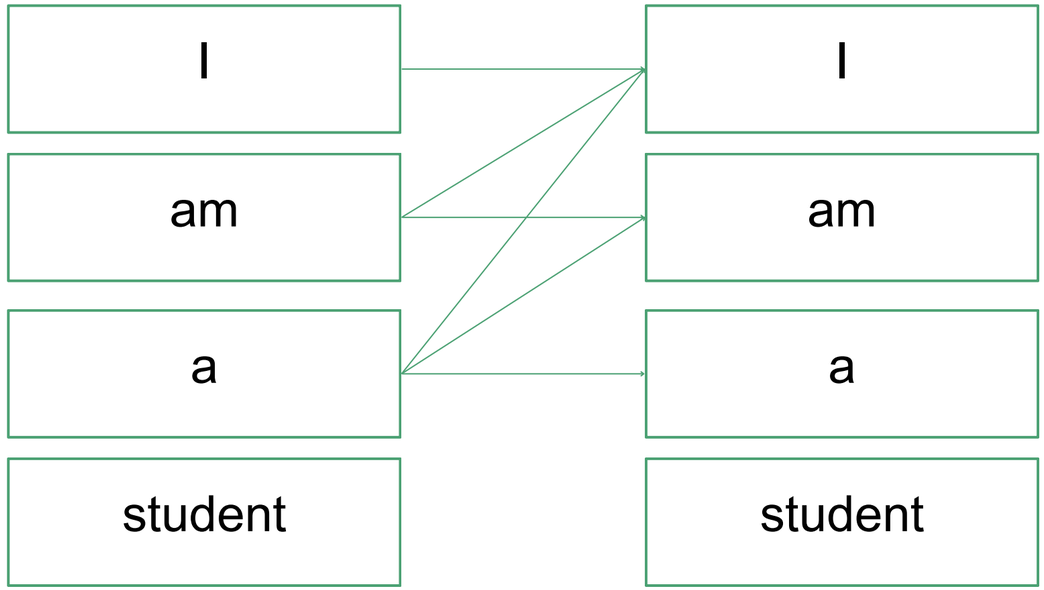


Decoder: Masked Multi-head attention

Very similar with multi-head attention in **encoder**

Purpose:

1. Find the focus of the sentence
2. Find the relationship between words and another previous words in a sentence

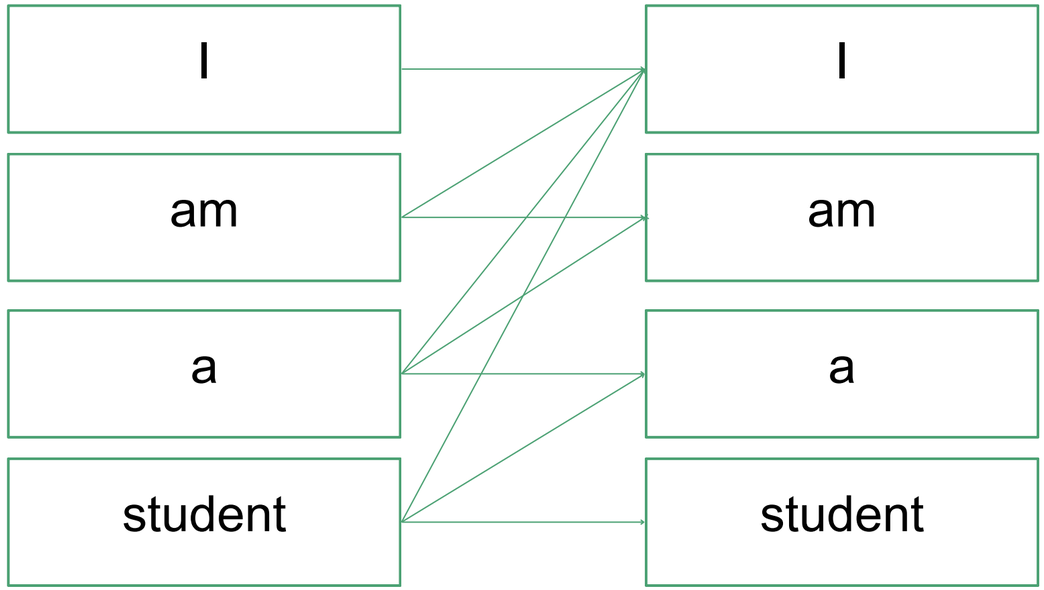


Decoder: Masked Multi-head attention

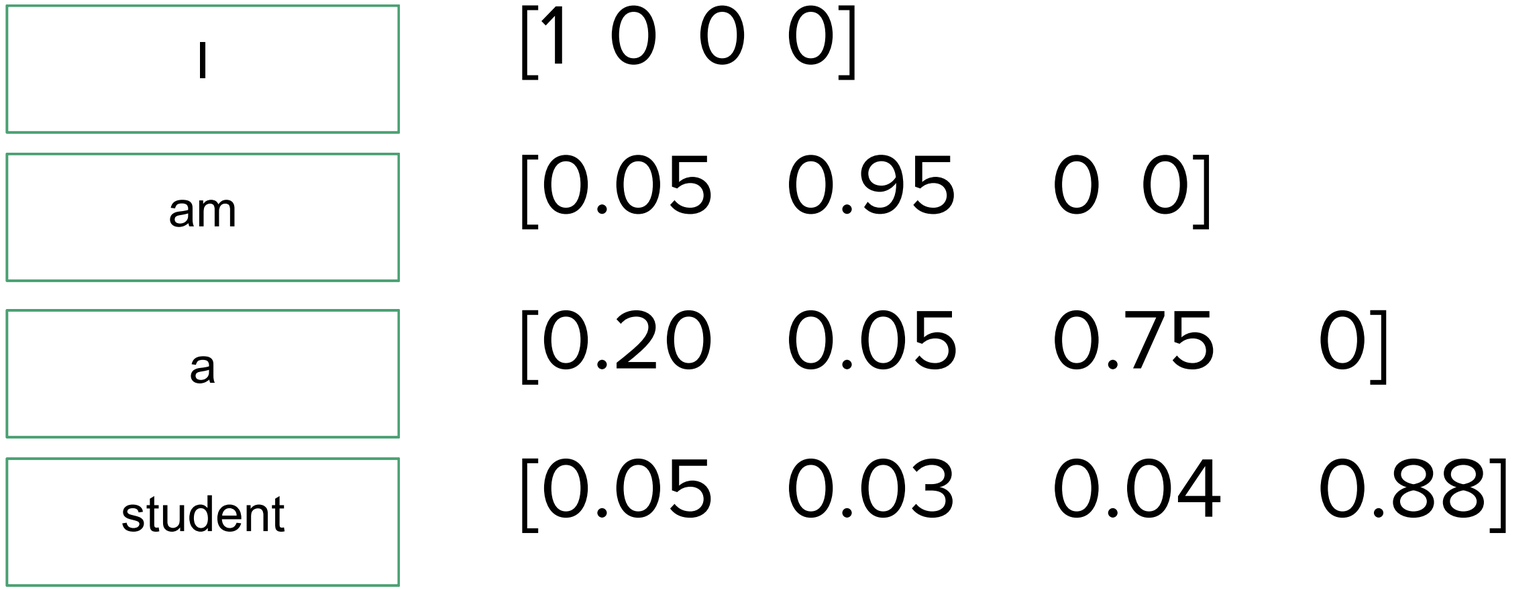
Very similar with multi-head attention in **encoder**

Purpose:

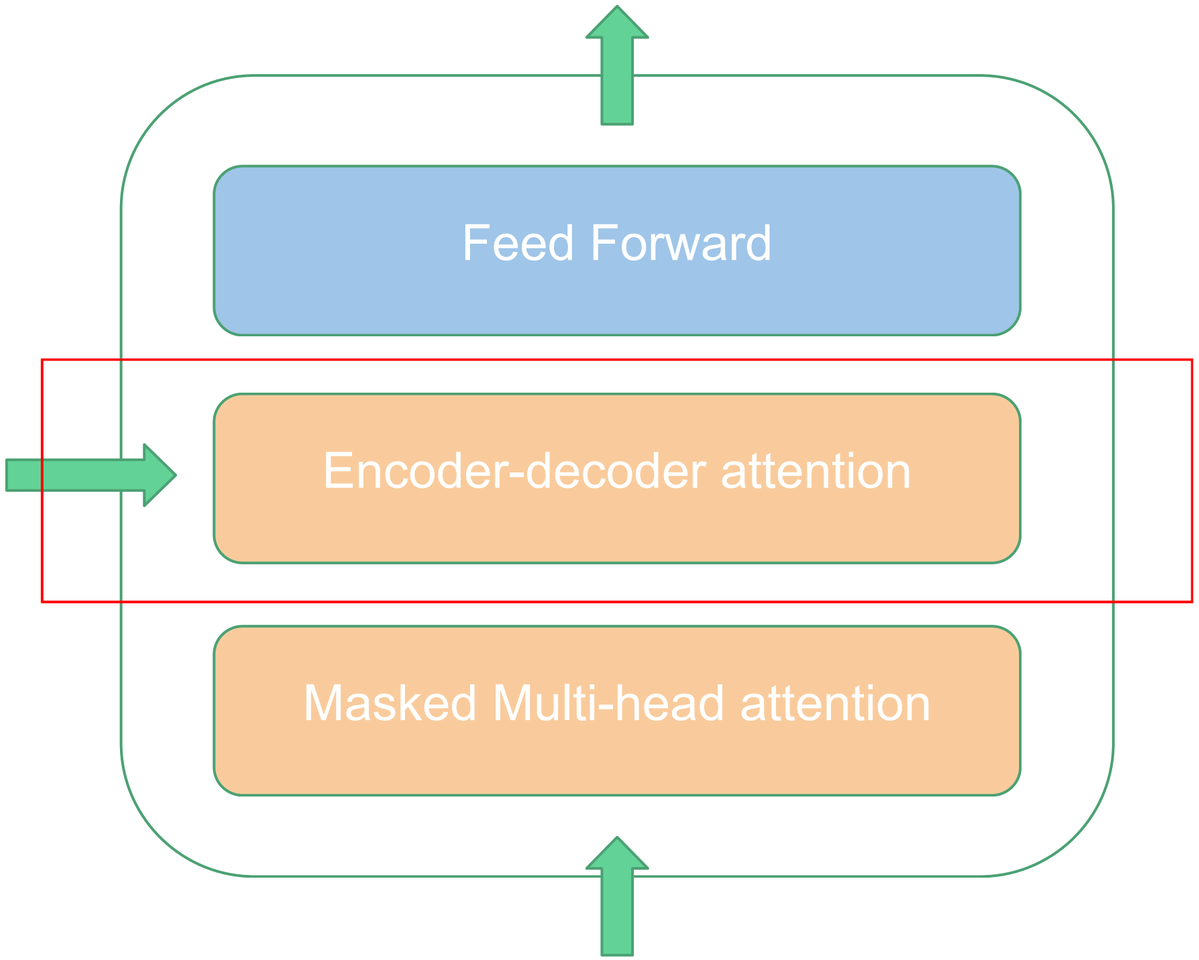
1. Find the focus of the sentence
2. Find the relationship between words and another previous words in a sentence



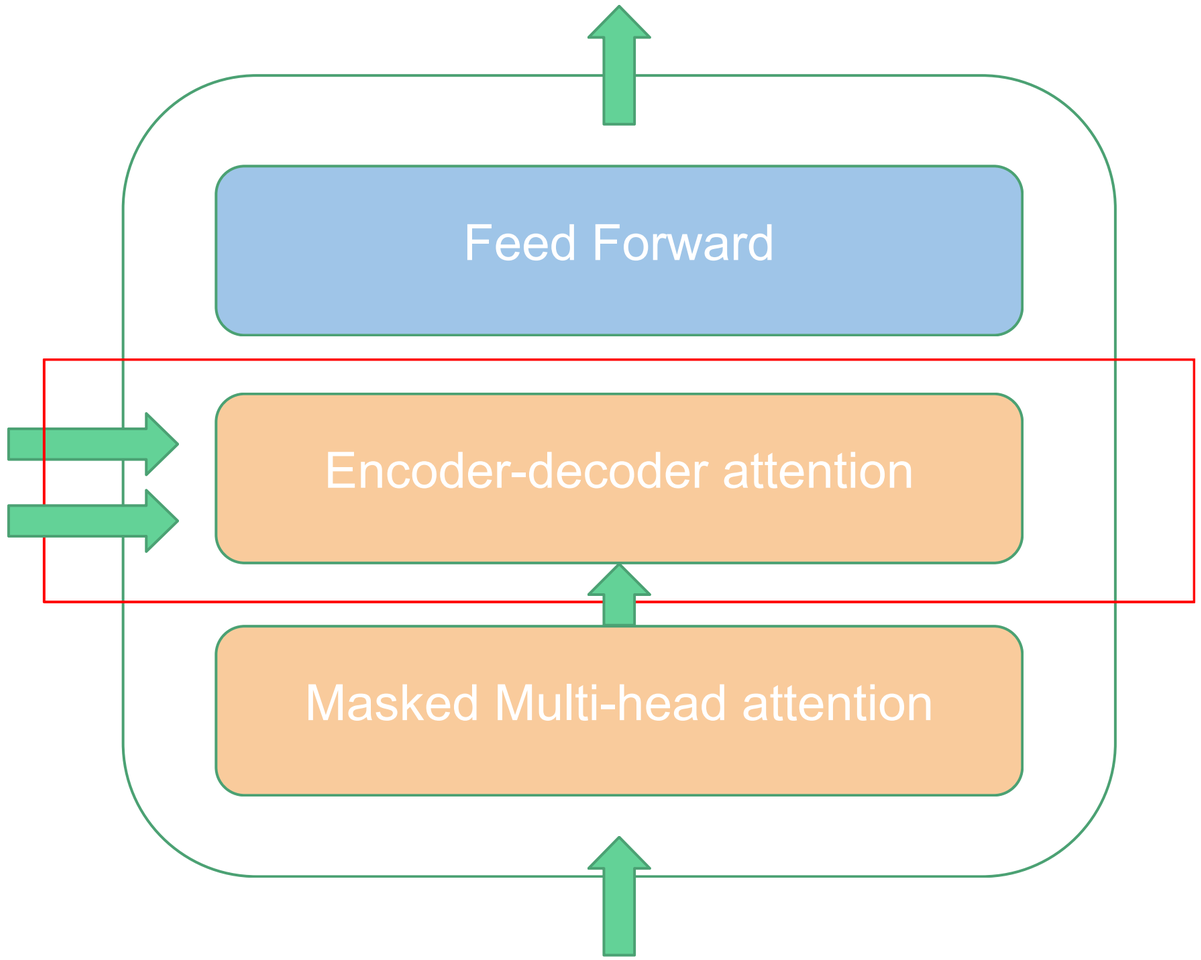
Decoder: Masked Multi-head attention



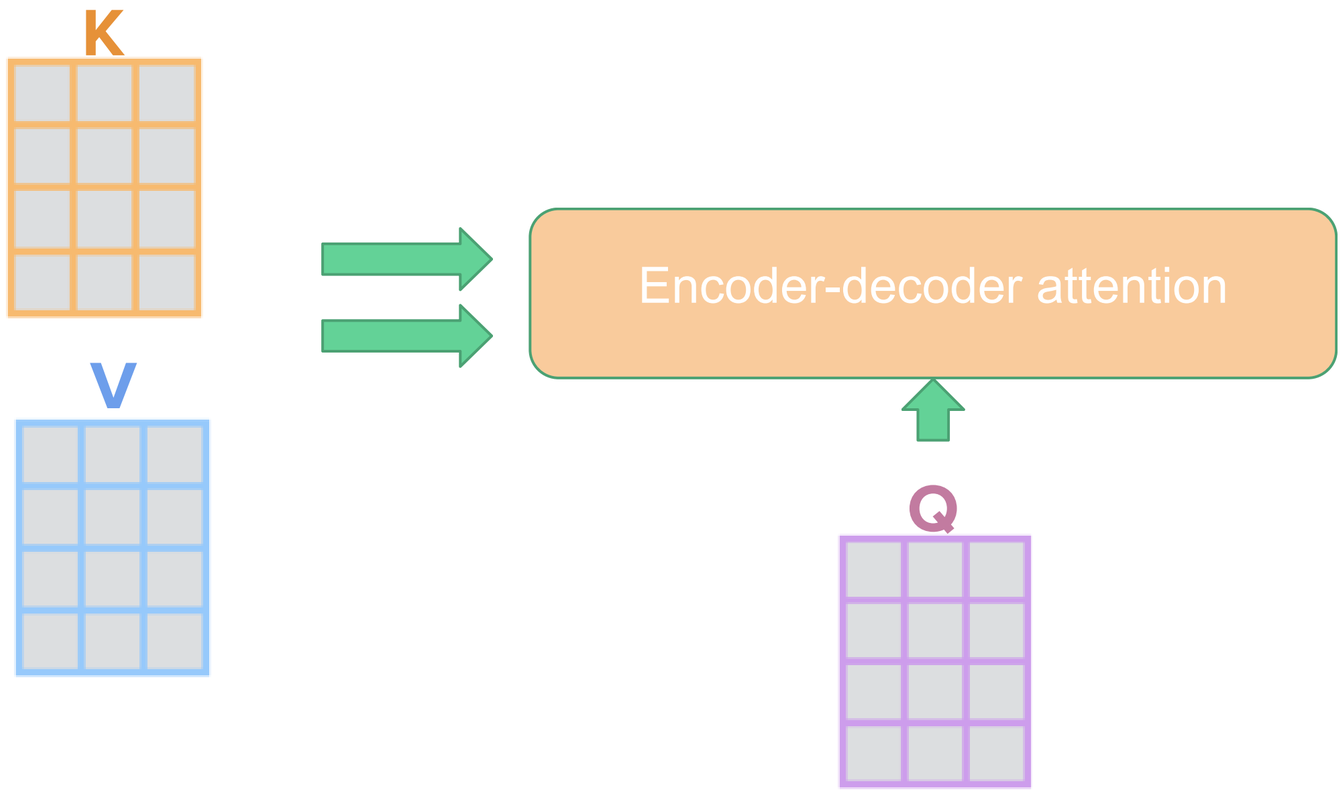
Decoder: Encoder-decoder attention



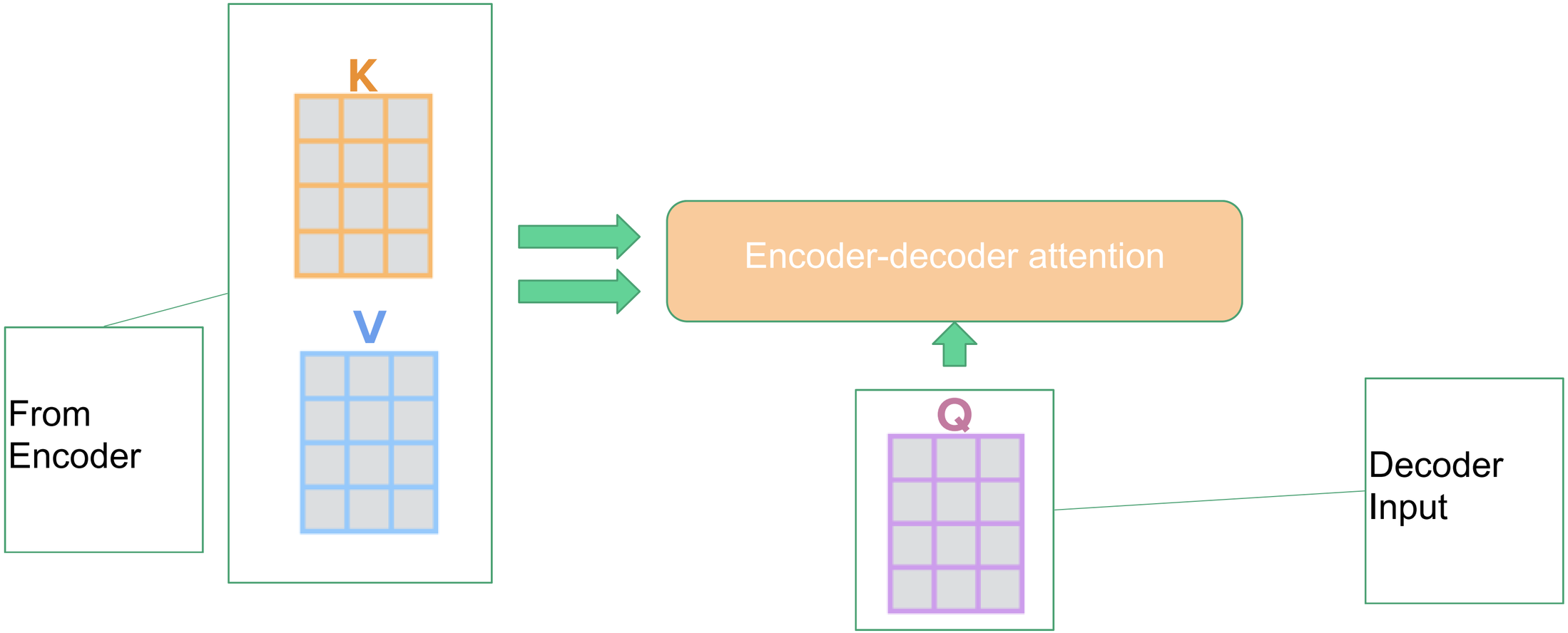
Decoder: Encoder-decoder attention



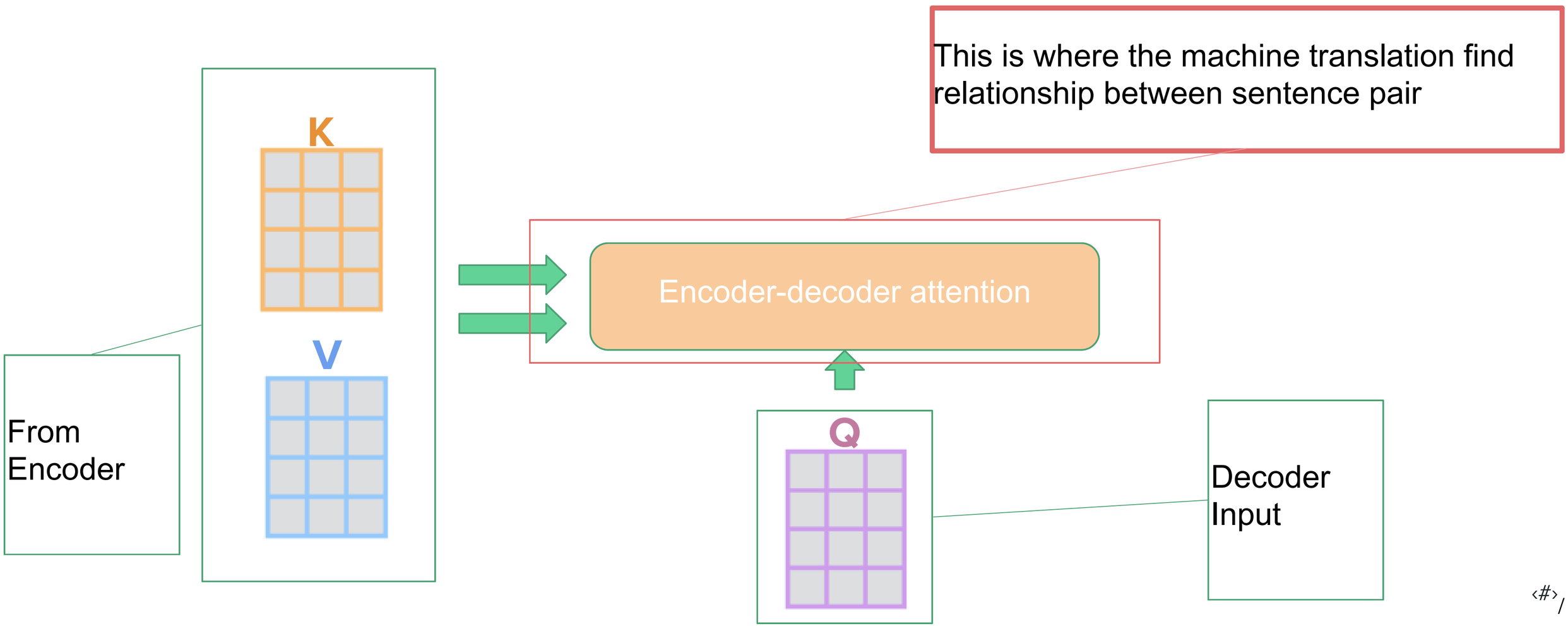
Decoder: Encoder-decoder attention



Decoder: Encoder-decoder attention



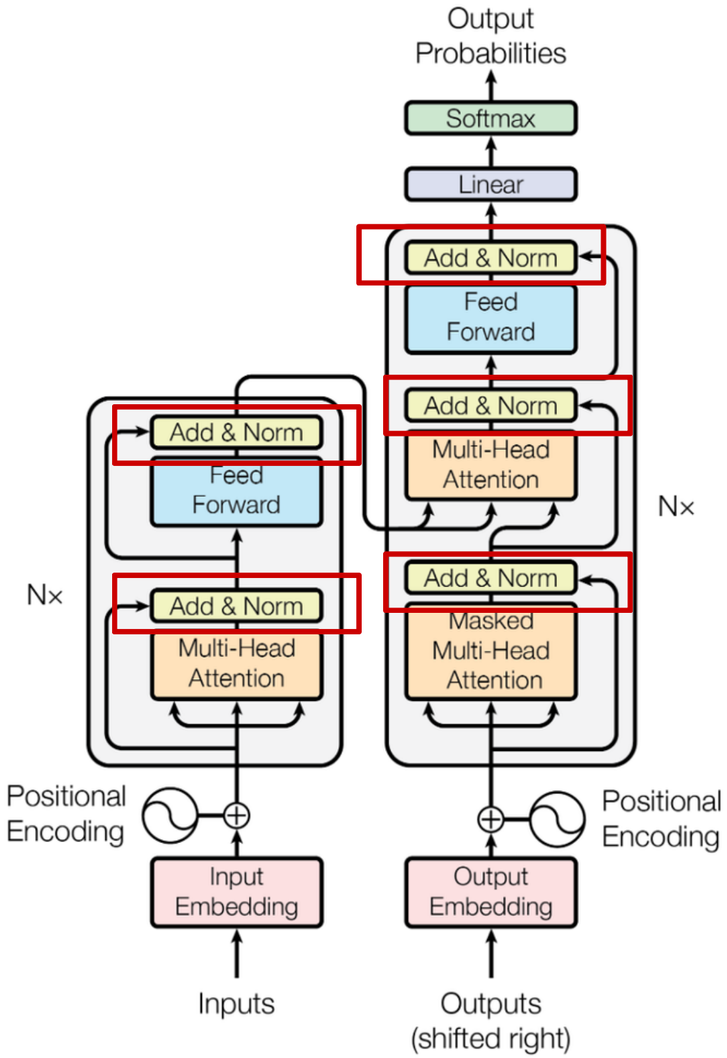
Decoder: Encoder-decoder attention



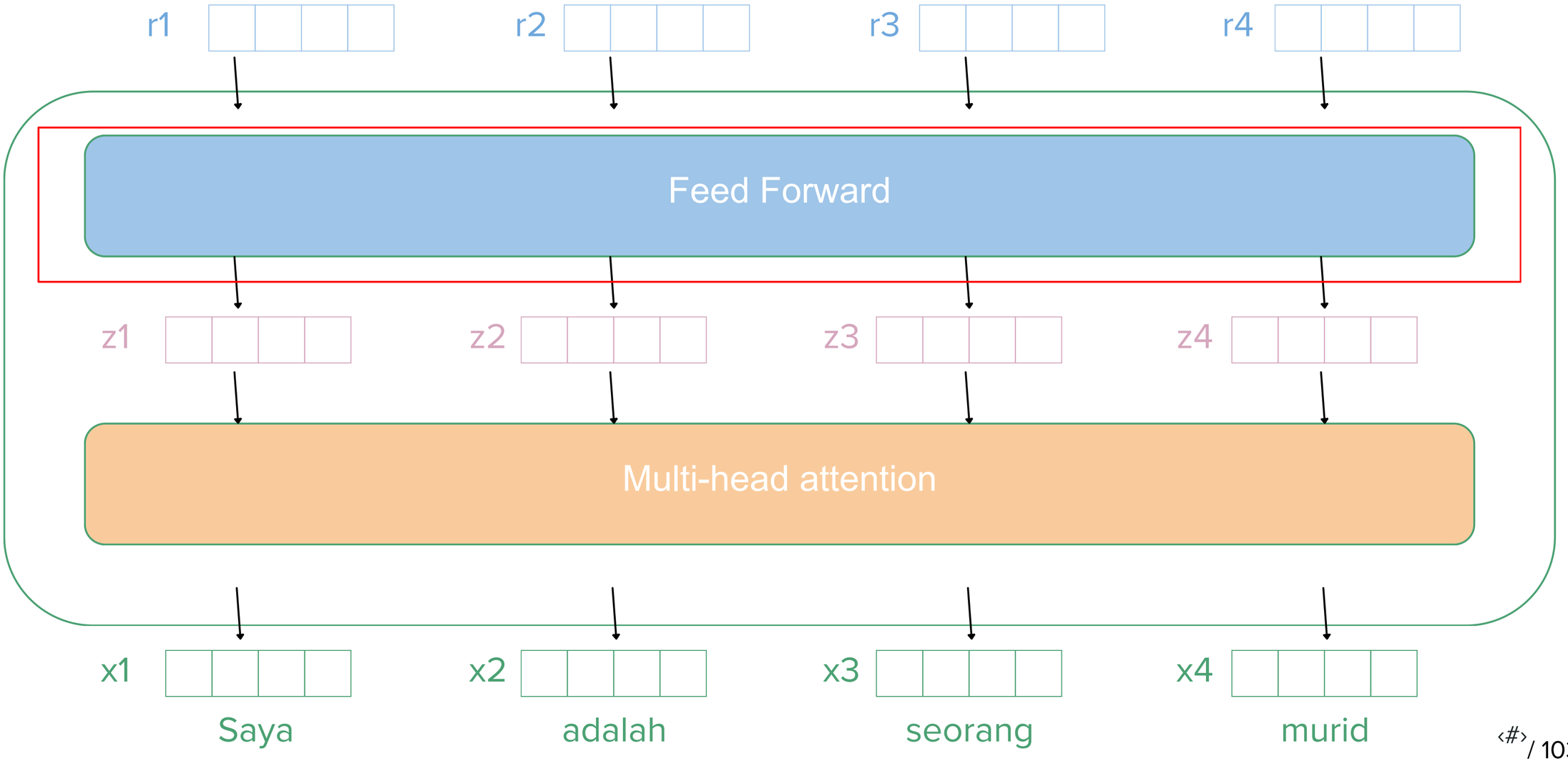
Any Question Guys ~

Transformers Add & Norm

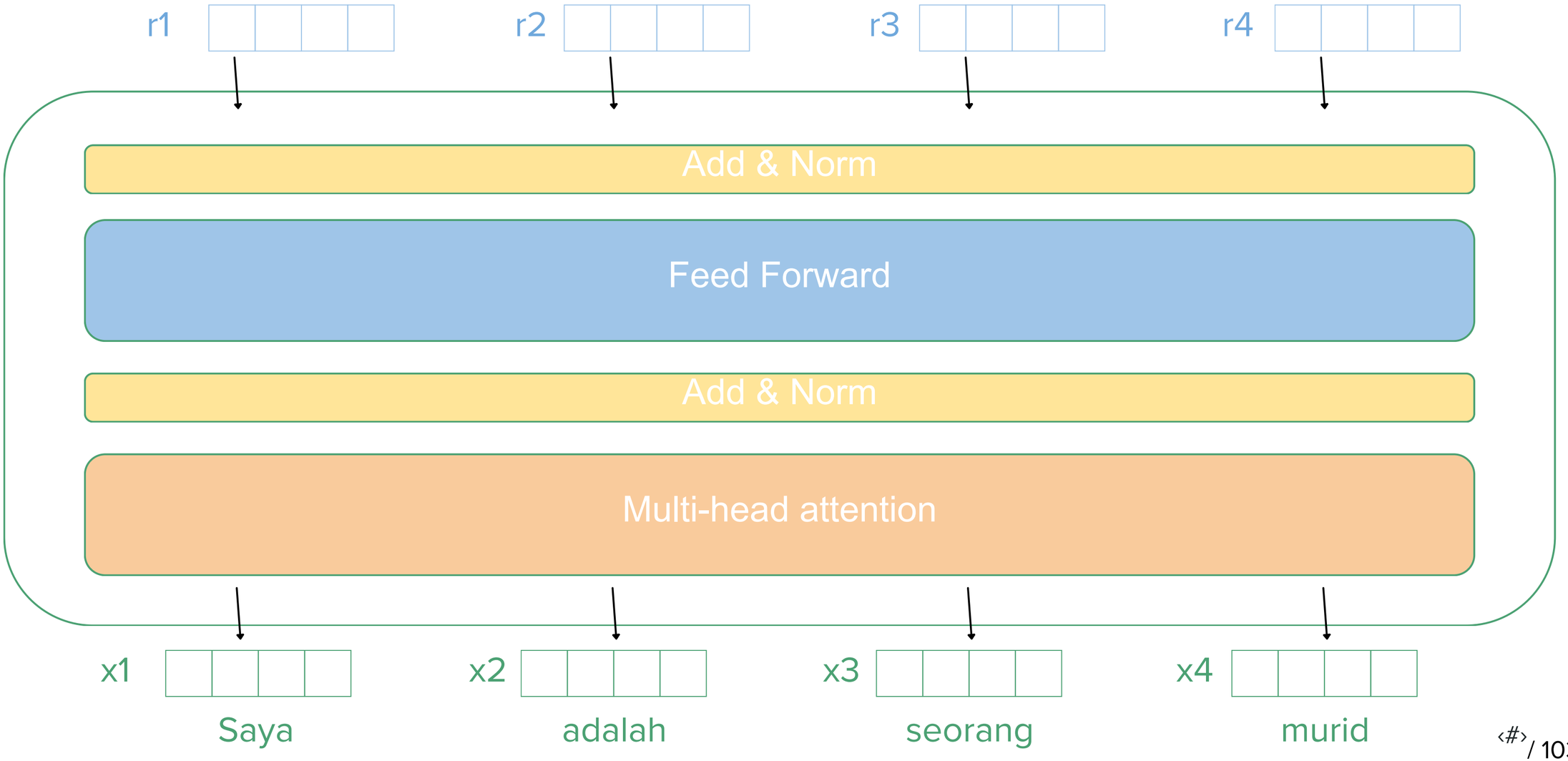
Add & Norm


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Add & Norm

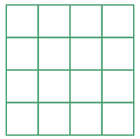

r1
r2
r3
r4
Feed Forward
z1
z2
z3
z4
Multi-head attention
x1
x2
x3
x4
Saya
adalah
seorang
murid

Add & Norm


r1
r2
r3
r4
Add & Norm
Feed Forward
Add & Norm
Multi-head attention
x1
x2
x3
x4
Saya
adalah
seorang
murid

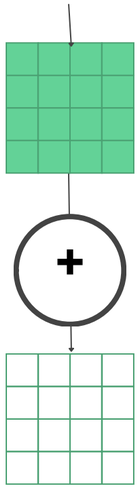
Add & Norm

Add & Norm

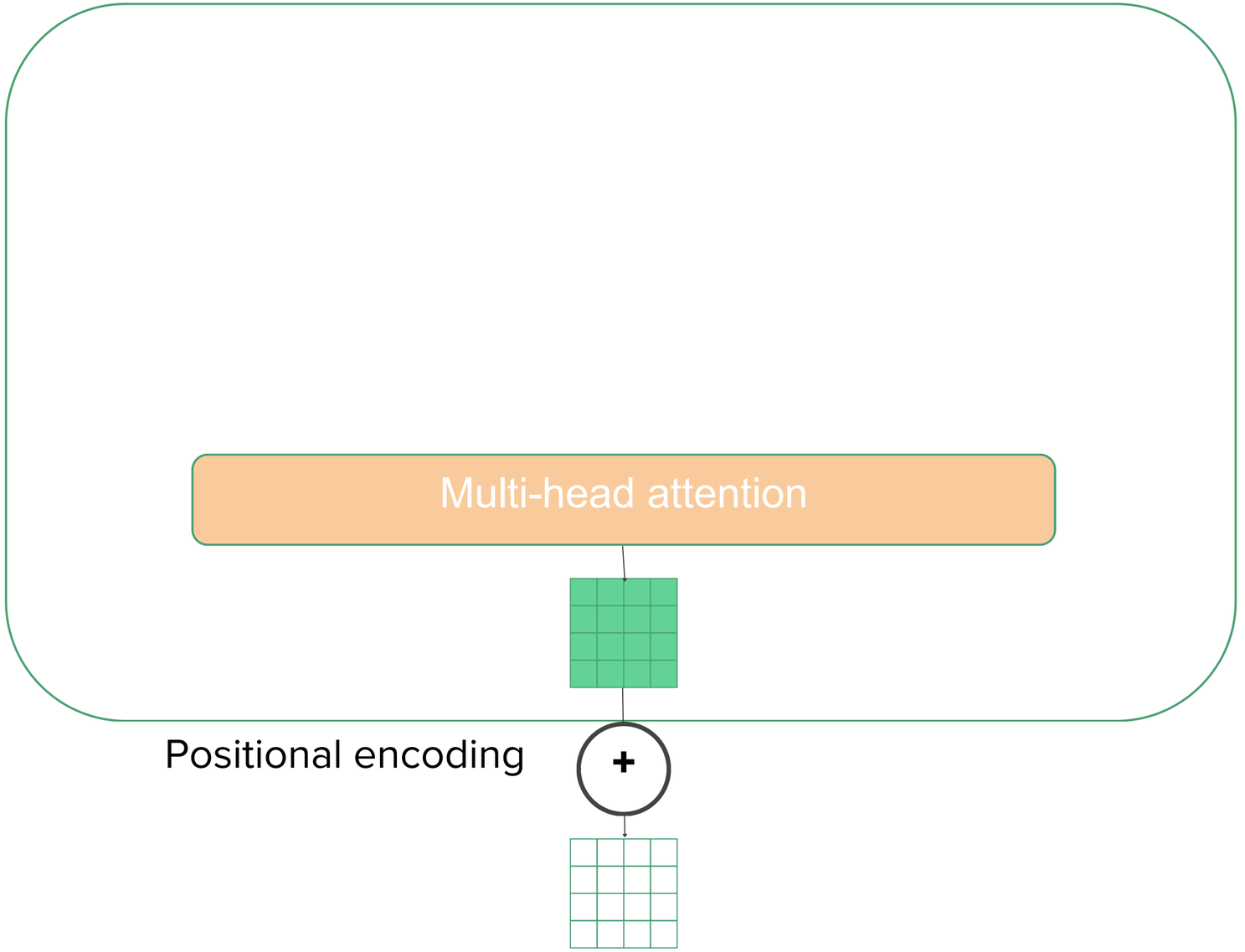


Add & Norm

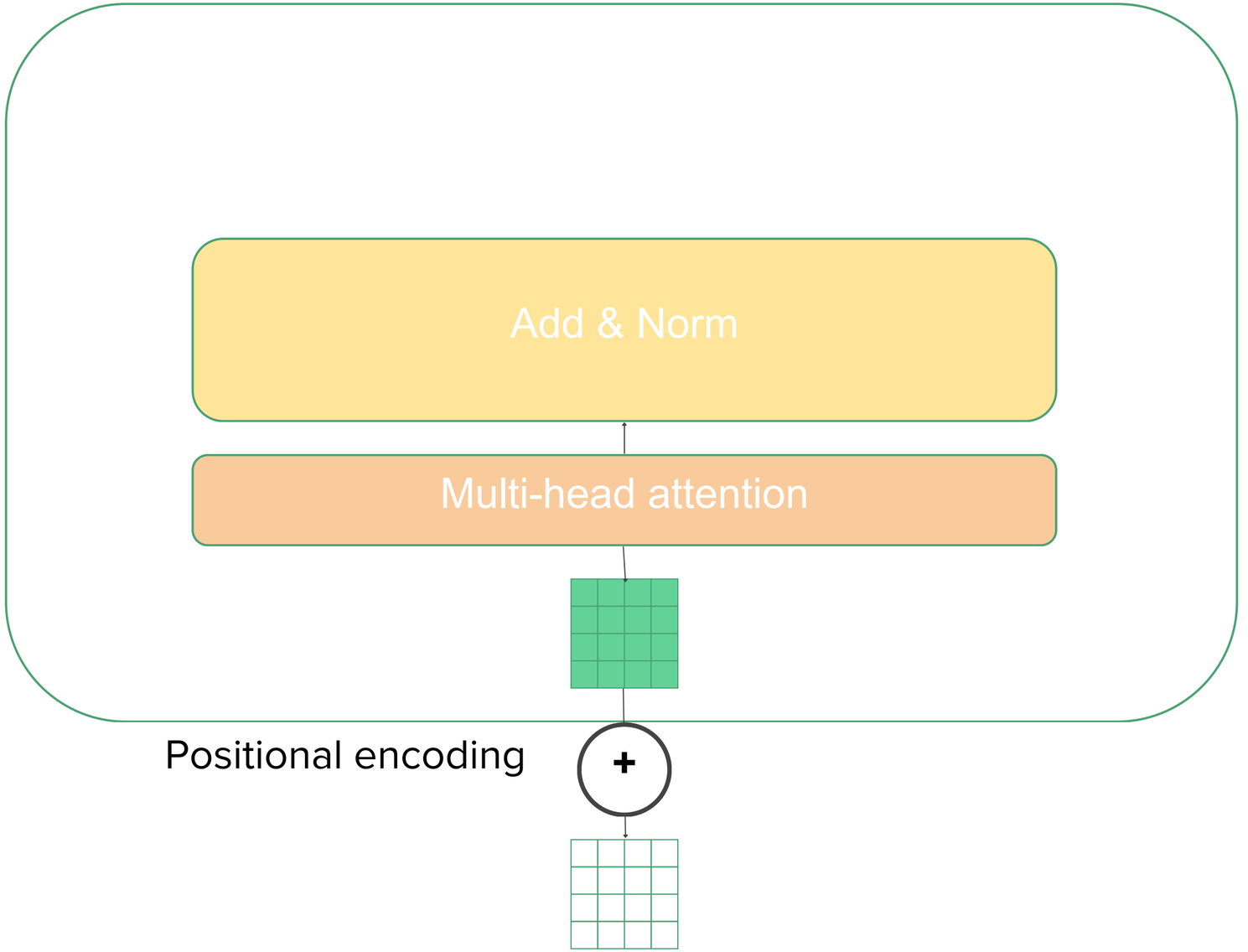
Positional encoding



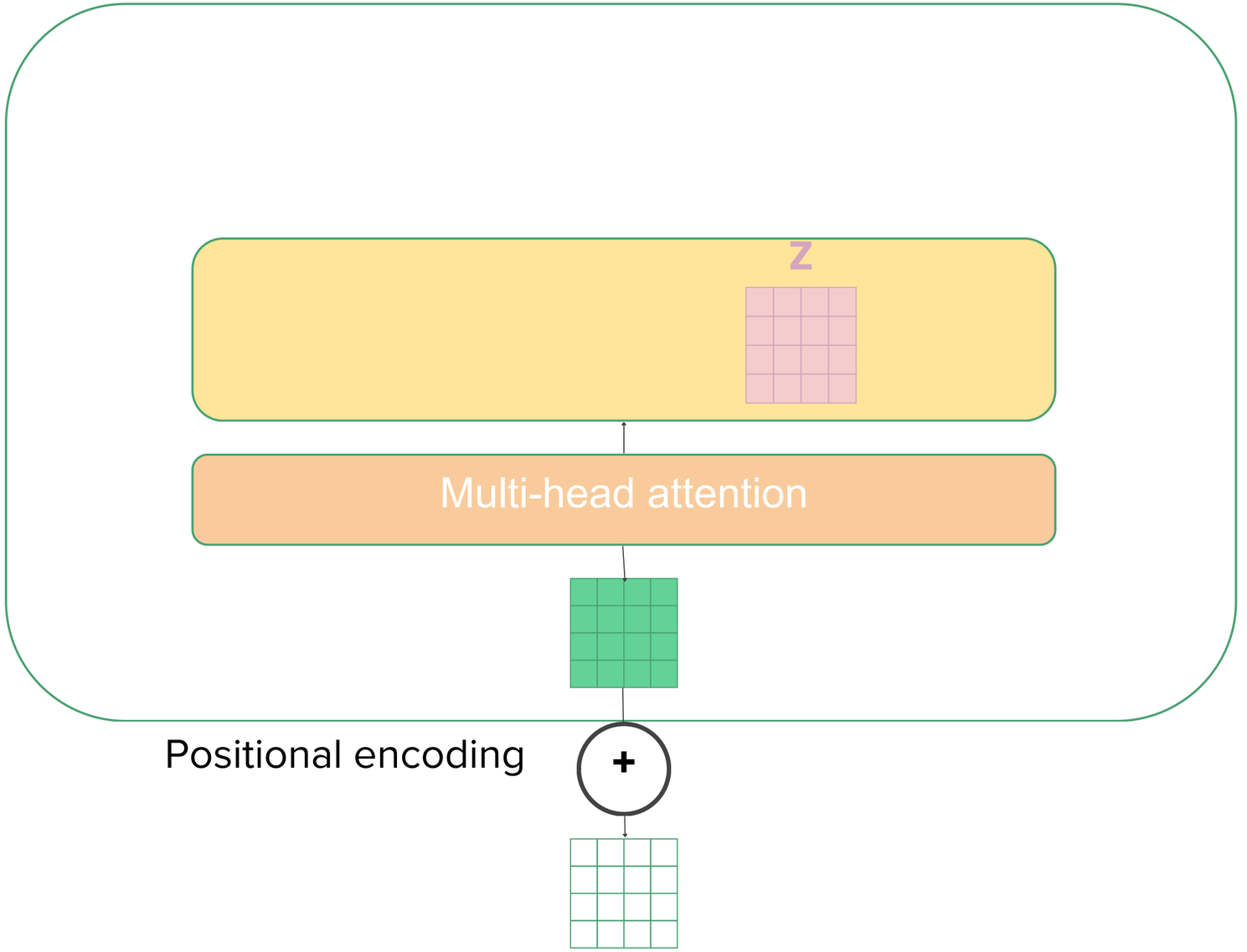
Add & Norm



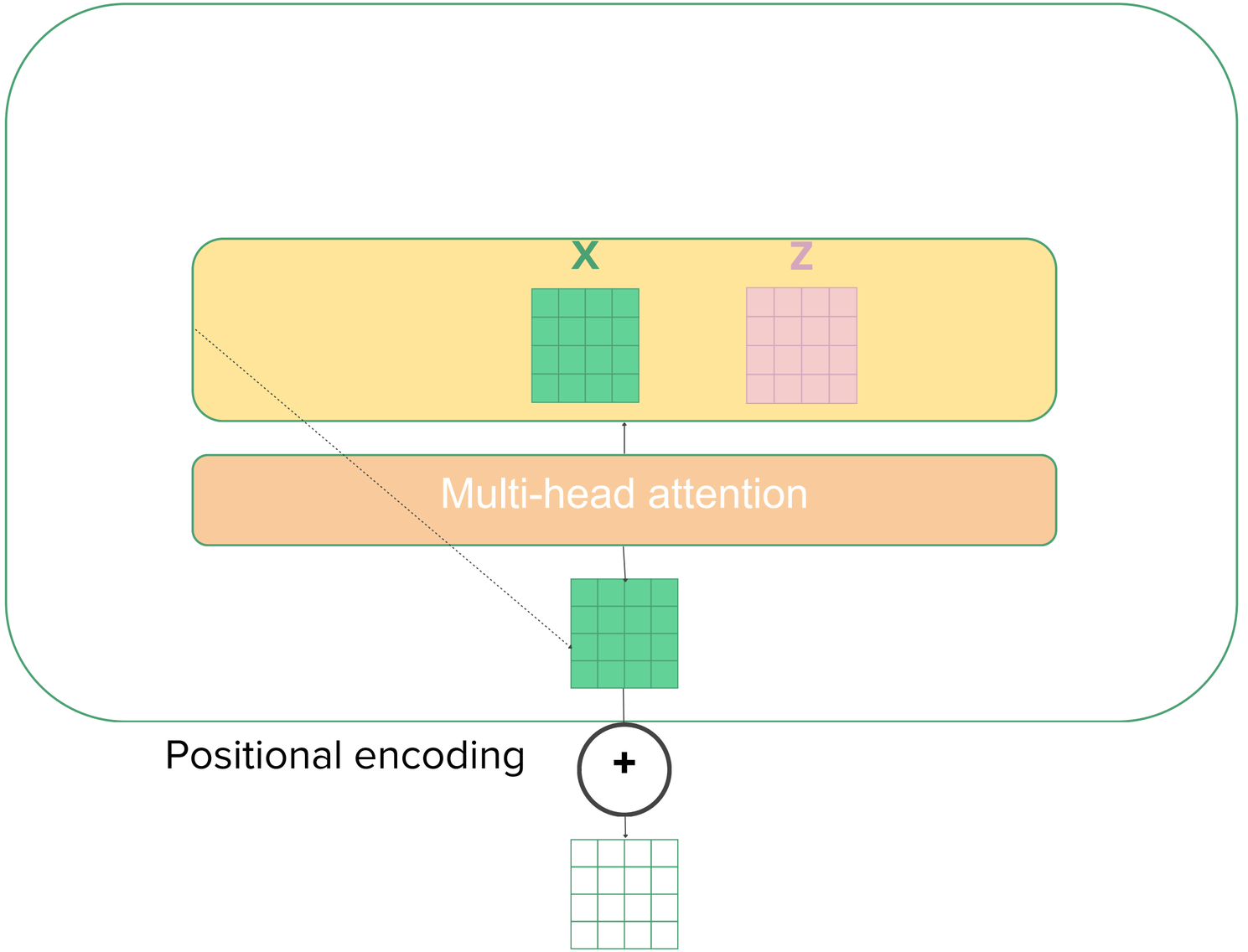
Add & Norm



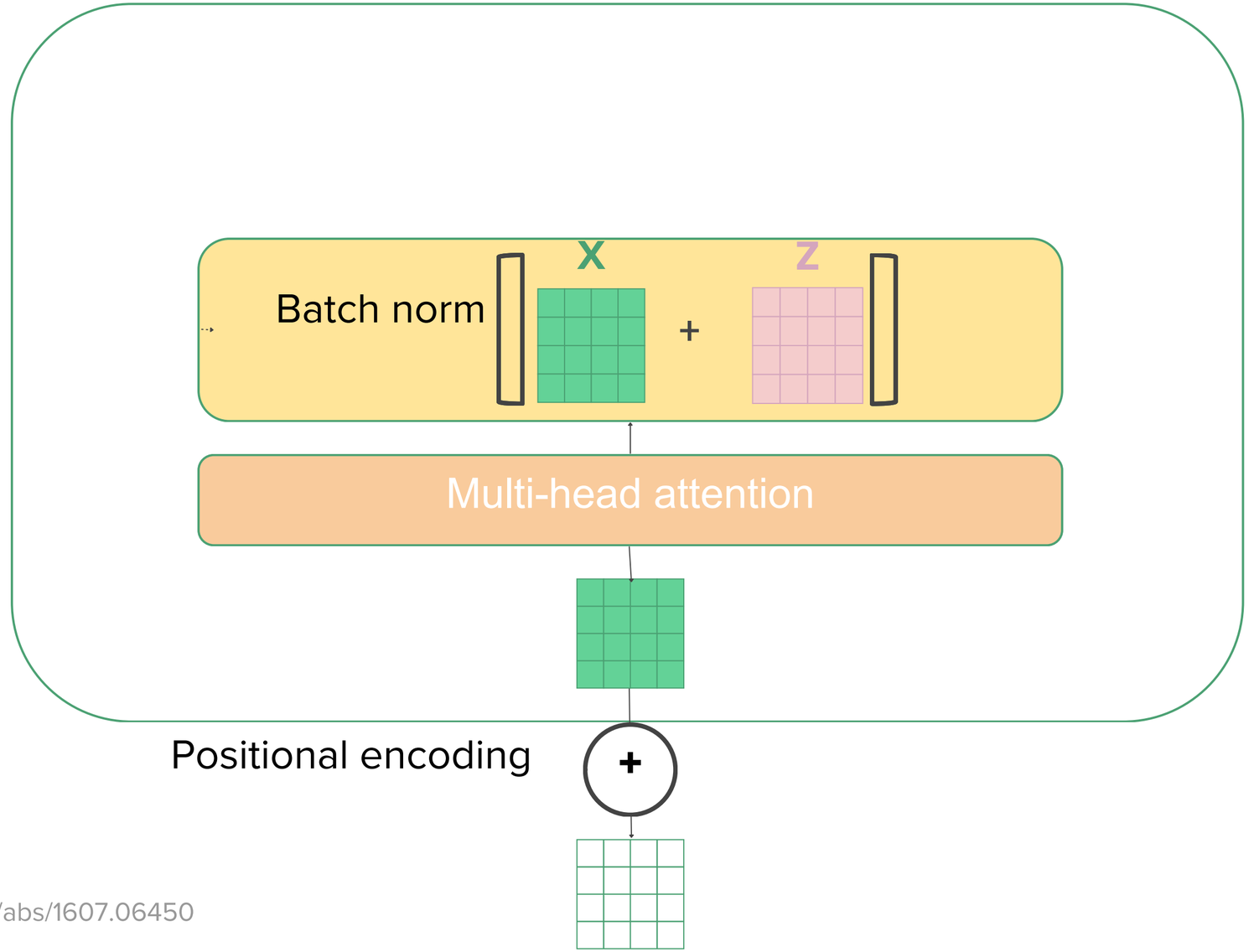
Add & Norm



Add & Norm

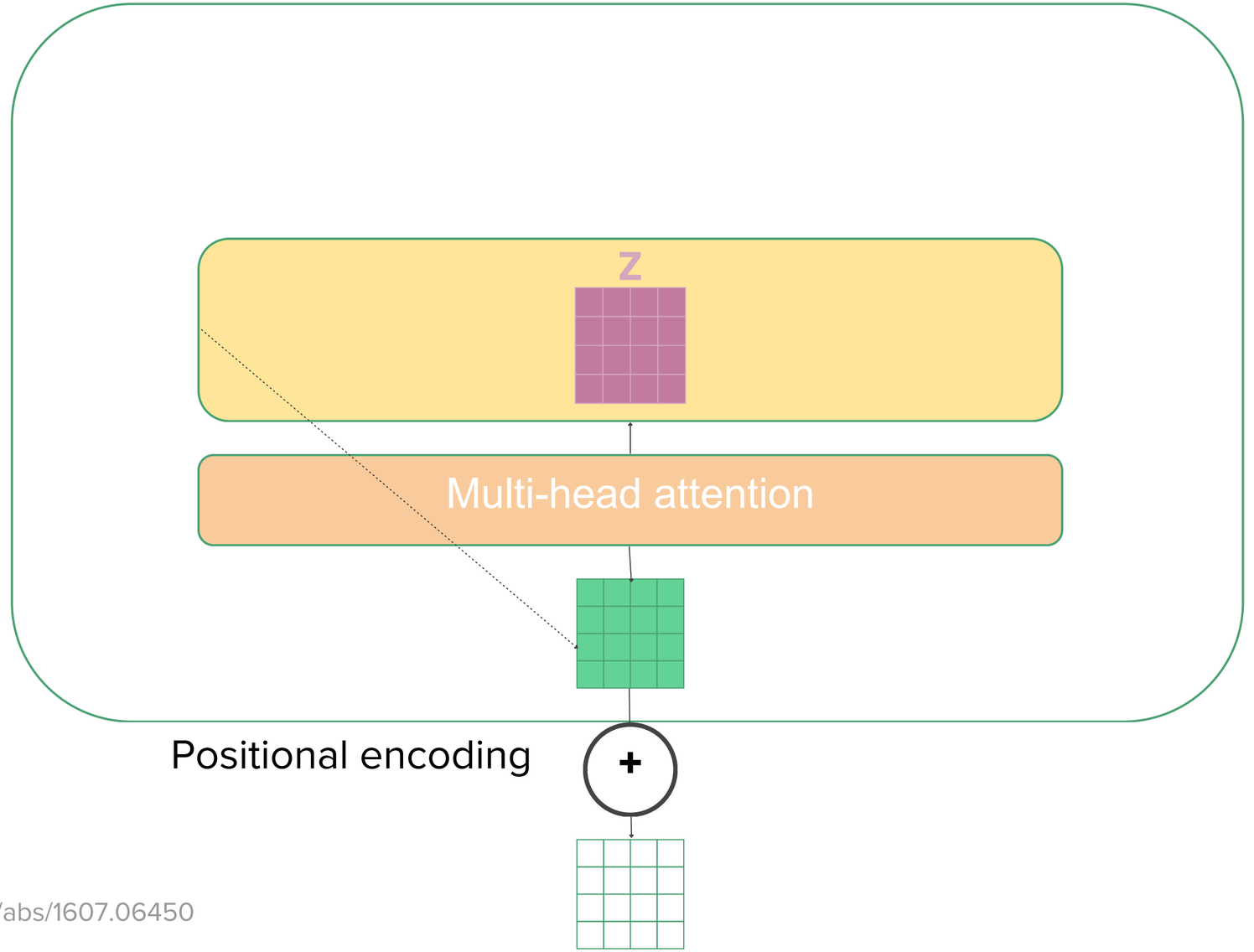


Add & Norm



Batch Normalization: https://arxiv.org/abs/1607.06450

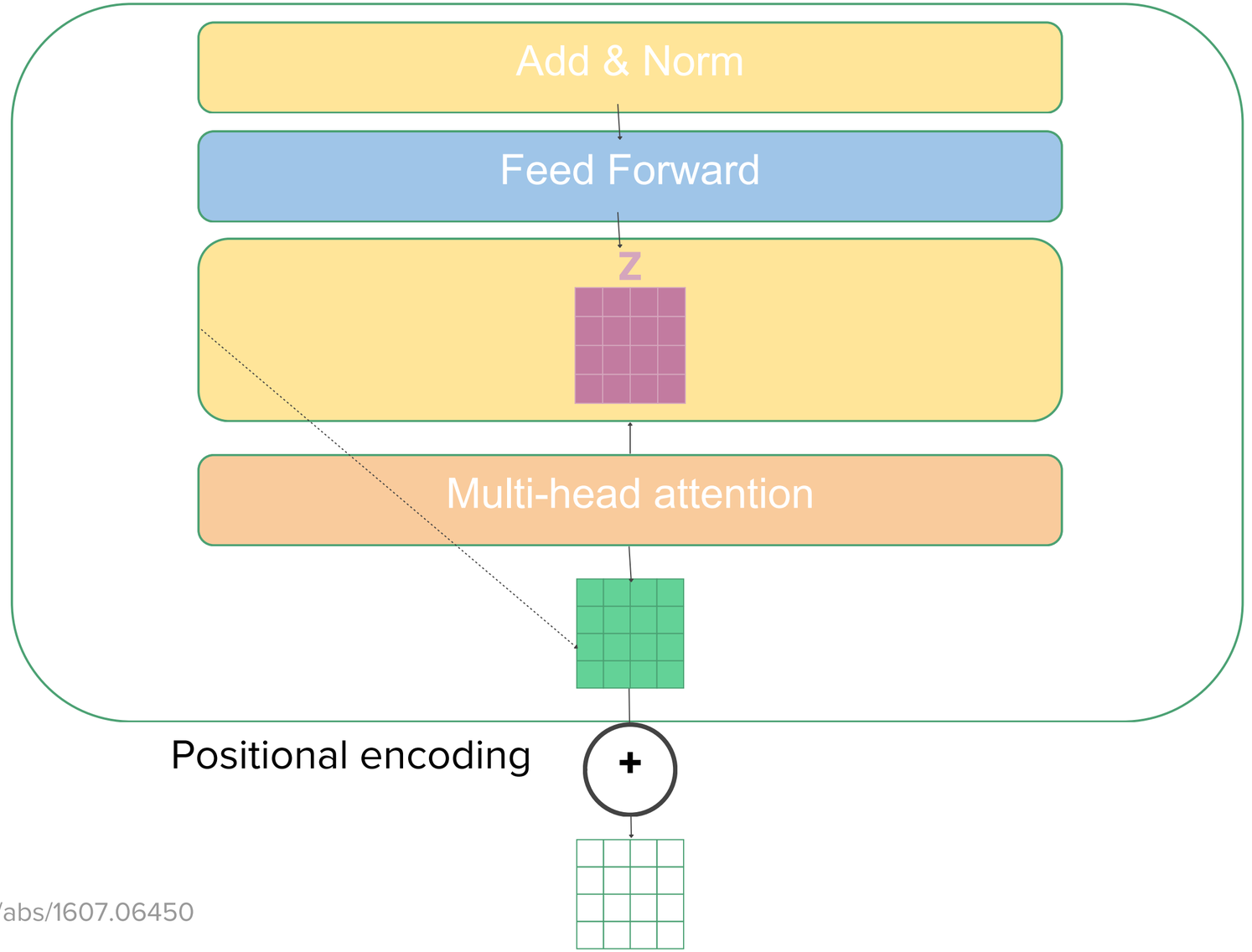
Add & Norm



Batch Normalization: https://arxiv.org/abs/1607.06450

https://arxiv.org/abs/1607.06450

Add & Norm



Batch Normalization: https://arxiv.org/abs/1607.06450

Any Question Guys ~

Transformers Performance

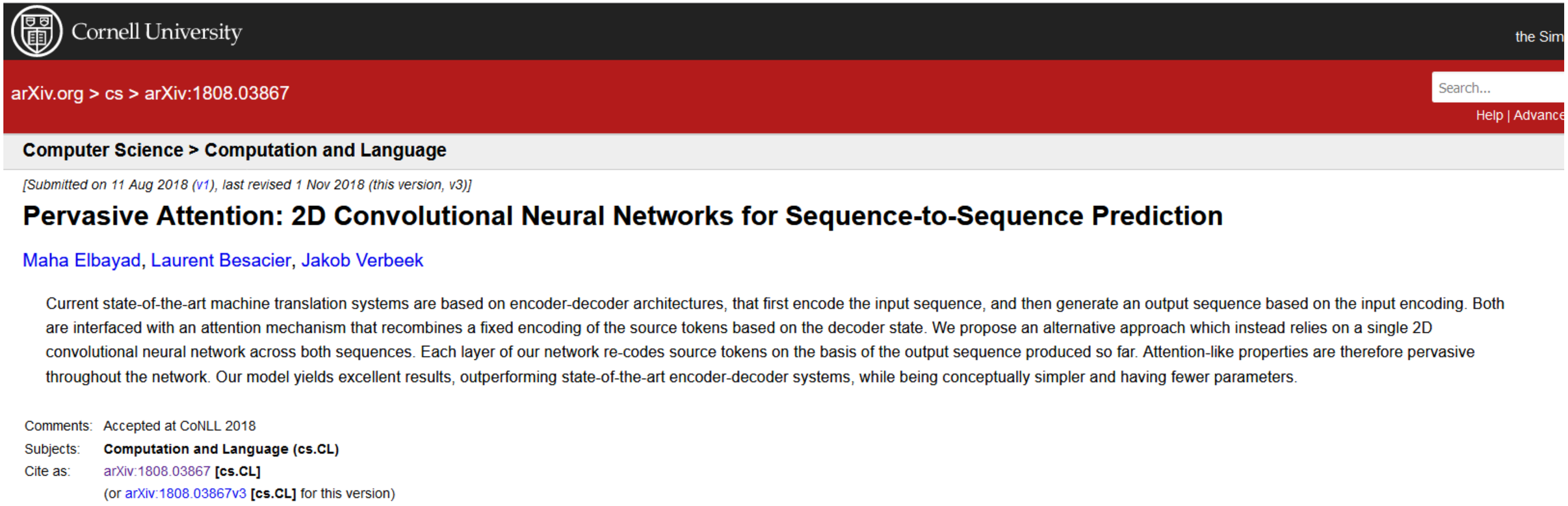
Transformers: Performance

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformers: Performance

But, https://arxiv.org/abs/1808.03867 performs better for seq2seq task



Cornell University

the Sim

arXiv.org > cs > arXiv:1808.03867

Search...

Help | Advance

Computer Science > Computation and Language

[Submitted on 11 Aug 2018 (v1), last revised 1 Nov 2018 (this version, v3)]

Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction

Maha Elbayad, Laurent Besacier, Jakob Verbeek

Current state-of-the-art machine translation systems are based on encoder-decoder architectures, that first encode the input sequence, and then generate an output sequence based on the input encoding. Both are interfaced with an attention mechanism that recombines a fixed encoding of the source tokens based on the decoder state. We propose an alternative approach which instead relies on a single 2D convolutional neural network across both sequences. Each layer of our network re-codes source tokens on the basis of the output sequence produced so far. Attention-like properties are therefore pervasive throughout the network. Our model yields excellent results, outperforming state-of-the-art encoder-decoder systems, while being conceptually simpler and having fewer parameters.

Comments: Accepted at CoNLL 2018
Subjects: **Computation and Language (cs.CL)**
Cite as: arXiv:1808.03867 **[cs.CL]**
(or arXiv:1808.03867v3 **[cs.CL]** for this version)

Transformers Development

Indonesia AI

Transformers-XL

https://arxiv.org/abs/1901.02860

Two new contribution from vanilla transformers:

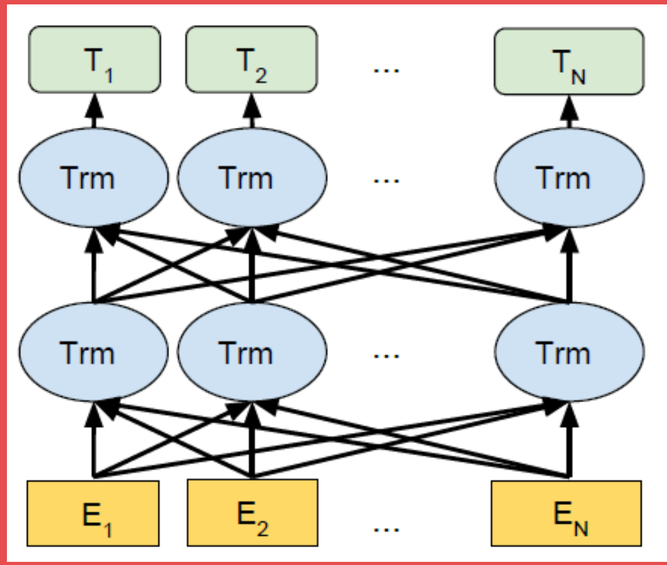
1. Recurrence mechanism
2. Relative positional embedding

BERT

Pre-training of Deep Bidirectional Transformers for Language Understanding

Main keys:

1. Masked Language Models (MLM)
2. Next Sentence Prediction



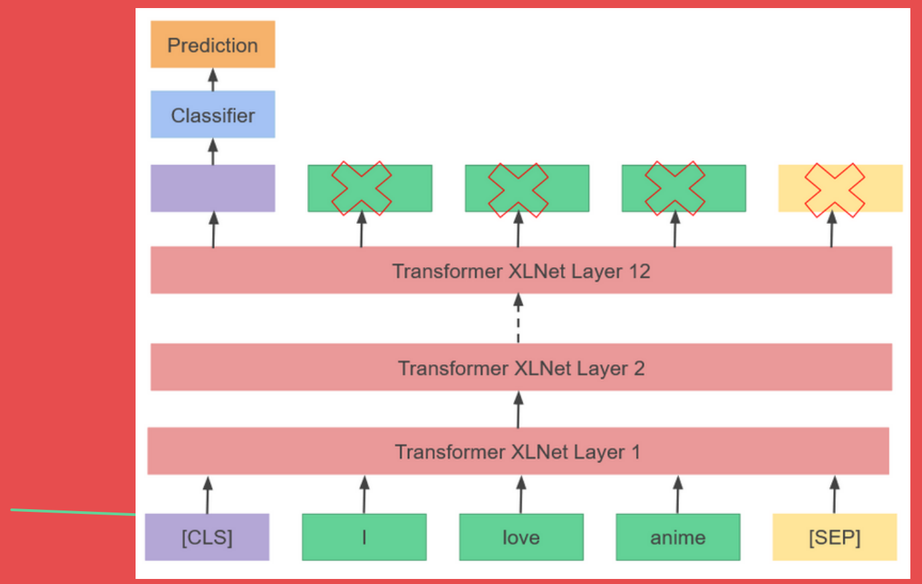
https://arxiv.org/abs/1810.04805

XL-Net

XLNet: Generalized Autoregressive Pretraining for Language Understanding

Main keys:

1. Permutation Language Modelling
2. Two-Stream Self-Attention



https://arxiv.org/abs/1906.08237

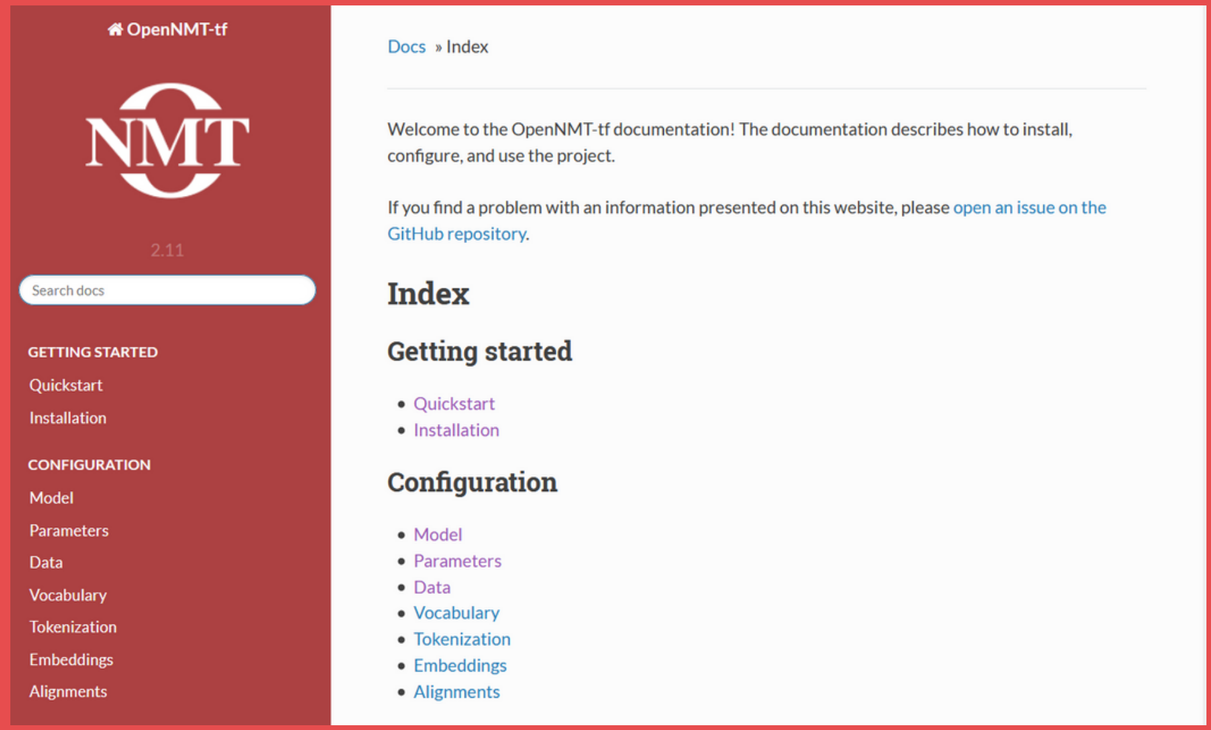
LET'S CODE!

Google Colaboratory
Machine translation using Transformers:
https://colab.research.google.com/drive/1CoT6zSiFy0IrCElFJyxeUwcu4peuPK_j?usp=sharing

Transformers from scratch:
https://www.tensorflow.org/tutorials/text/transformer

Experiment

- We will use OpenNMT library
- 10.000 English-German sentence pair



https://opennmt.net/OpenNMT-tf/index.html

Reference

1. Paper: https://arxiv.org/abs/1706.03762
2. Transformers illustration: http://jalammar.github.io/illustrated-transformer/
3. https://www.youtube.com/watch?v=TQQIZhbC5ps
4. https://www.youtube.com/watch?v=KN3ZL65Dze0&t=133s
5. https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Any Question Guys ~

Terimakasih!