Transformers : From moderation to code generation

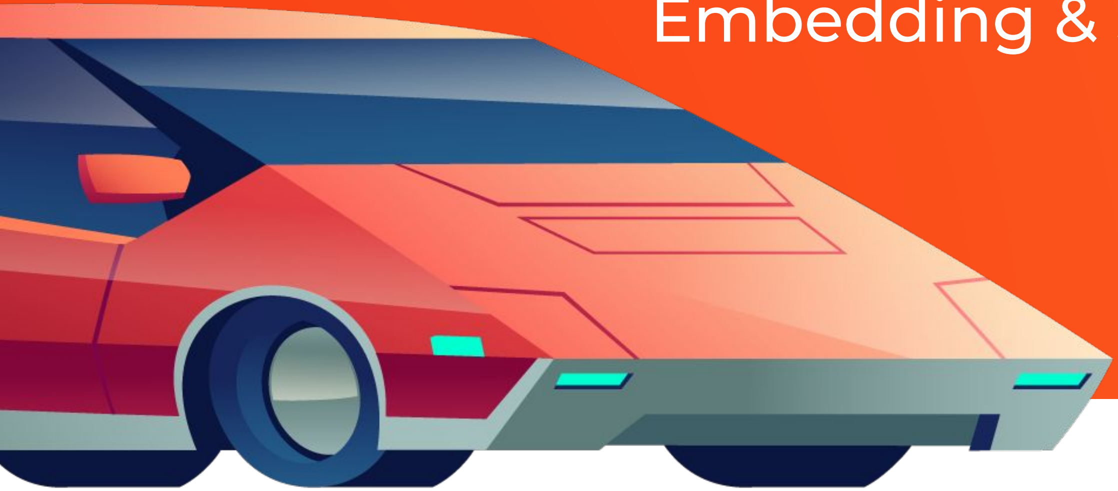
Pierre GUILLAUME
pierre.guillaume@epita.fr

Corentin DUCHÊNE
corentin.duchene@epita.fr

Introduction

Social media
content moderation

GitHub Copilot

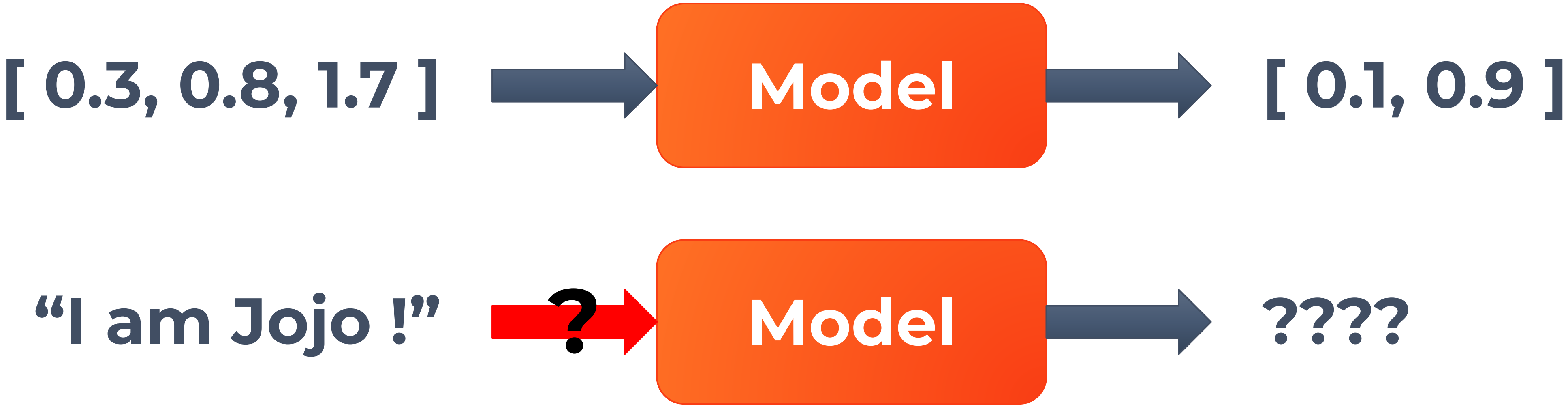


I

NLP essentials

Embedding & Self-supervised Learning

Using words as input to the model ?!



[0.3, 0.8, 1.7]

Model

[0.1, 0.9]

“I am Jojo !”

?

Model

????

Using words as input to the model ?!

“I am Jojo !” → “i am <name>” → [“i”, “am”, …]

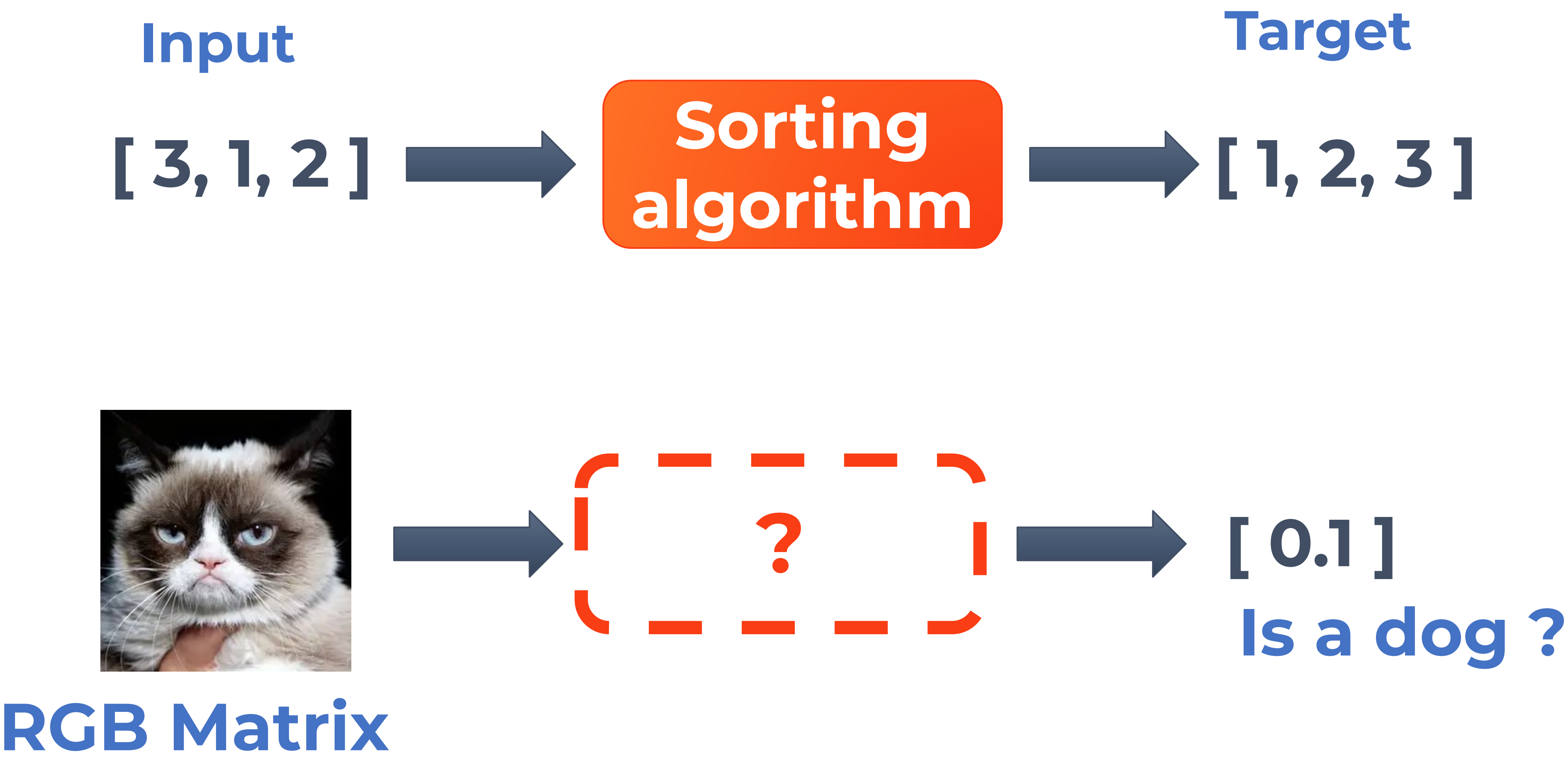
“i” → [0.18, 0.62, 0.12]

“am” → [0.56, 0.27, 0.09]

…

“I am Jojo !” → [[0.18, 0.62, 0.12], …]

Algorithmic solution (without machine learning)



Input

[3, 1, 2]

Sorting algorithm

Target

[1, 2, 3]

?

[0.1]

Is a dog ?

RGB Matrix

Different types of machine learning

Supervised

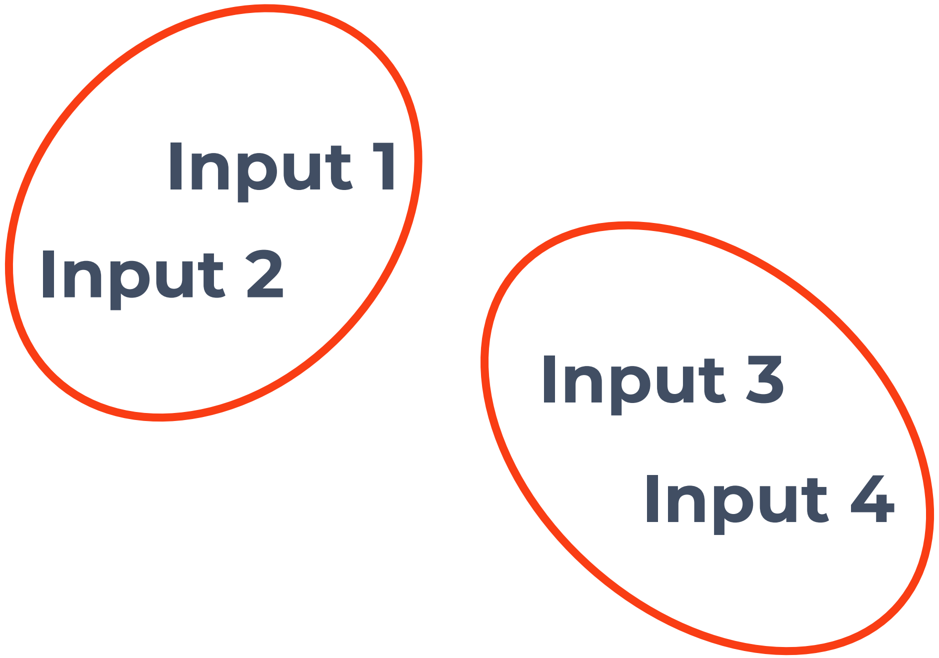
Input 1 → Target 1

Input 2 → Target 2

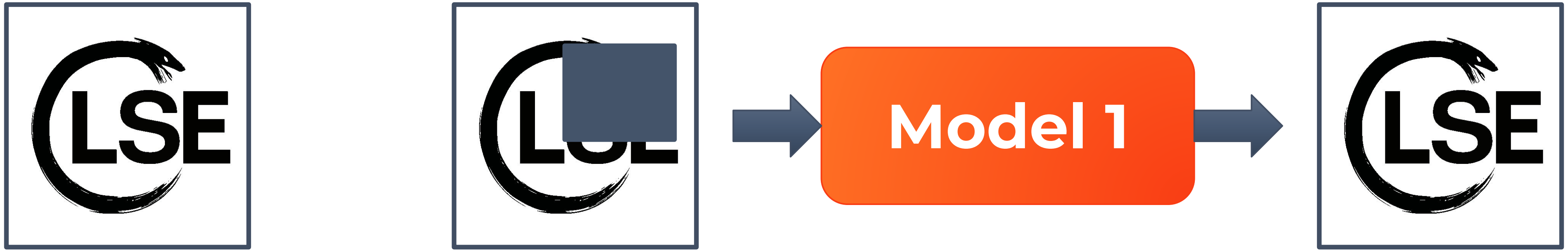
Input 3 → Target 3

Input 4 → Target 4

Unsupervised

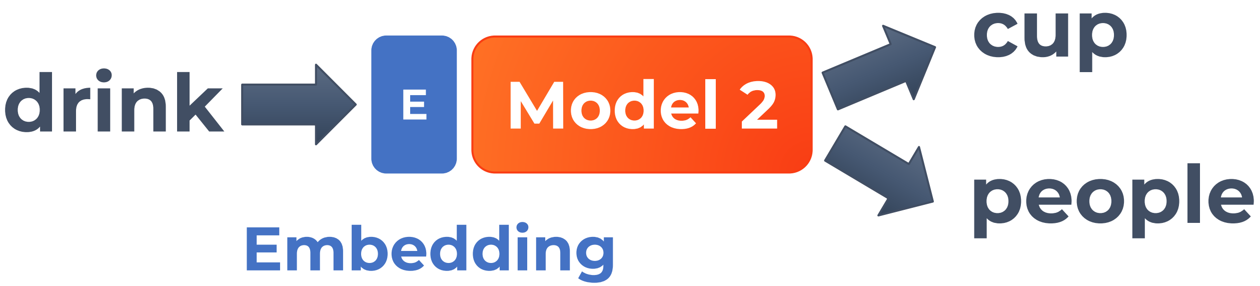


Self-supervised learning



Model 1

"People drink a cup of coffee"



drink

E

Model 2

cup

people

Embedding

Word Embedding

Not contextual

- Word2Vect
- Glove
- FastText

Contextual

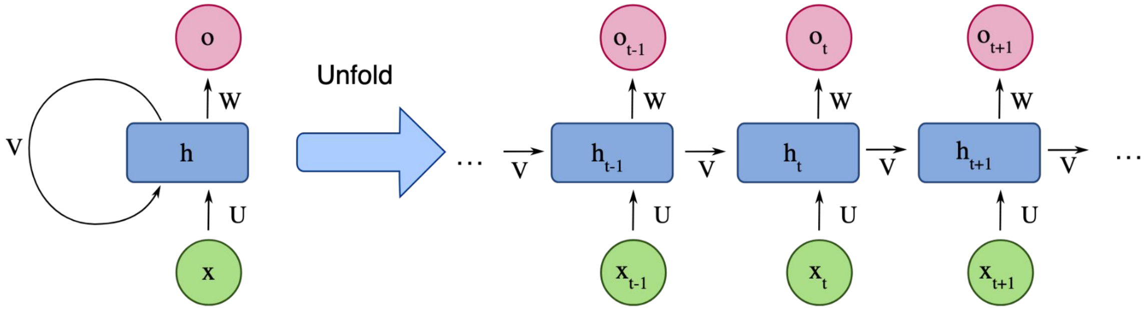
- ELMo
- BERT
- CoVe

II

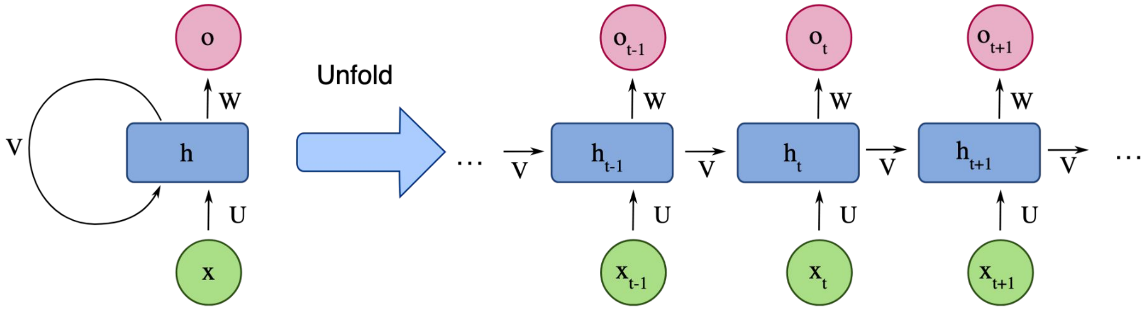
From RNN to Transformers

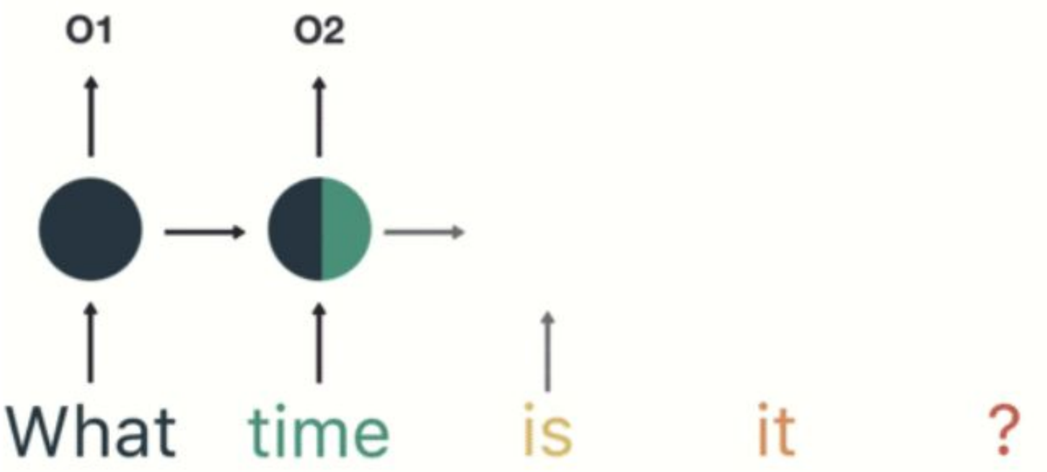
The state of the art before the transformers

From RNN to Transformers

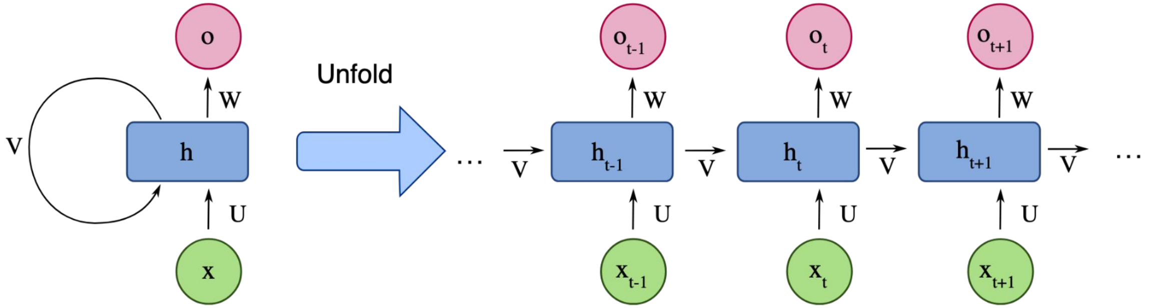


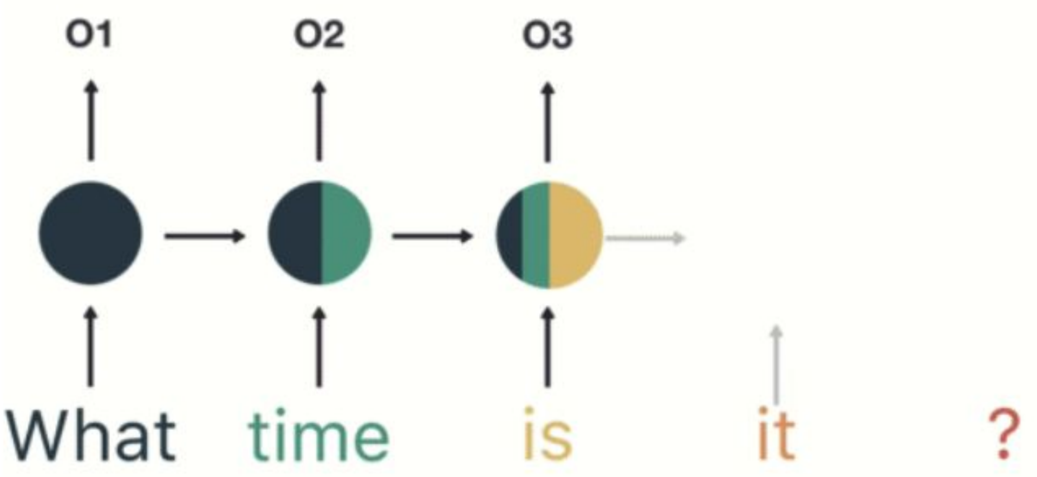
From RNN to Transformers



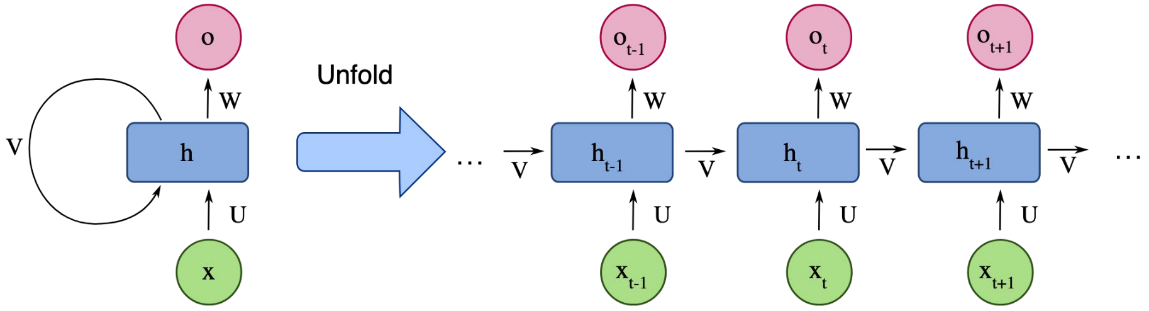


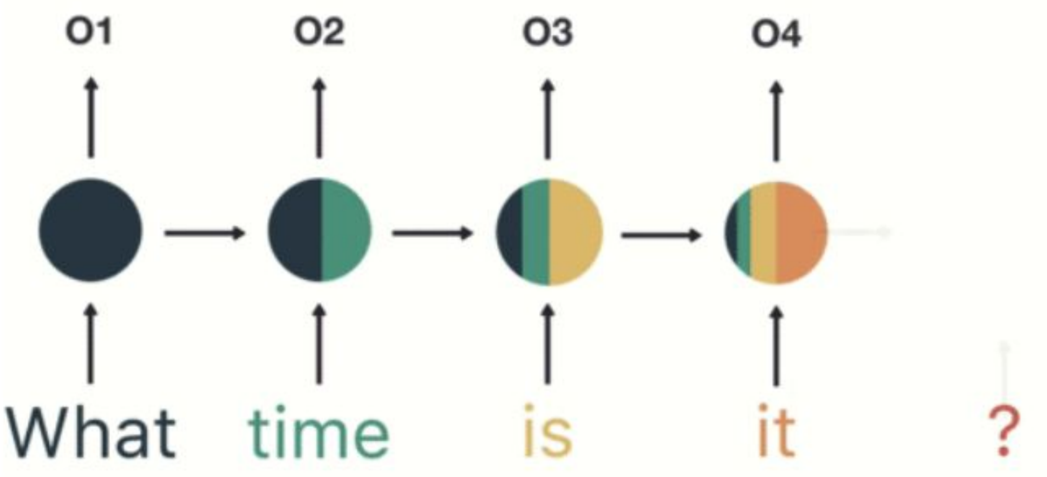
From RNN to Transformers



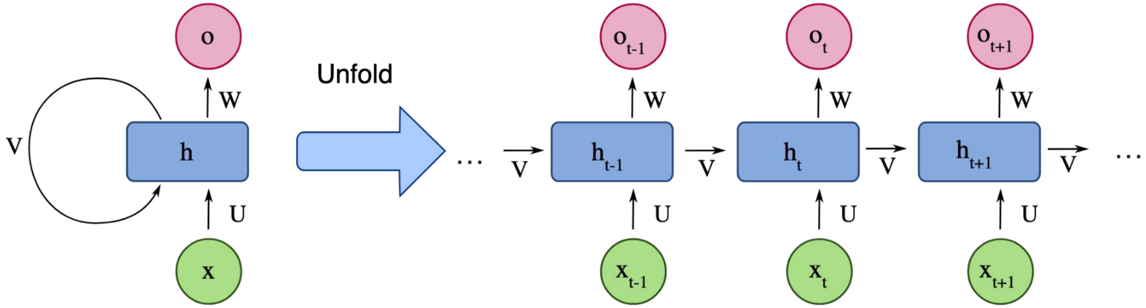


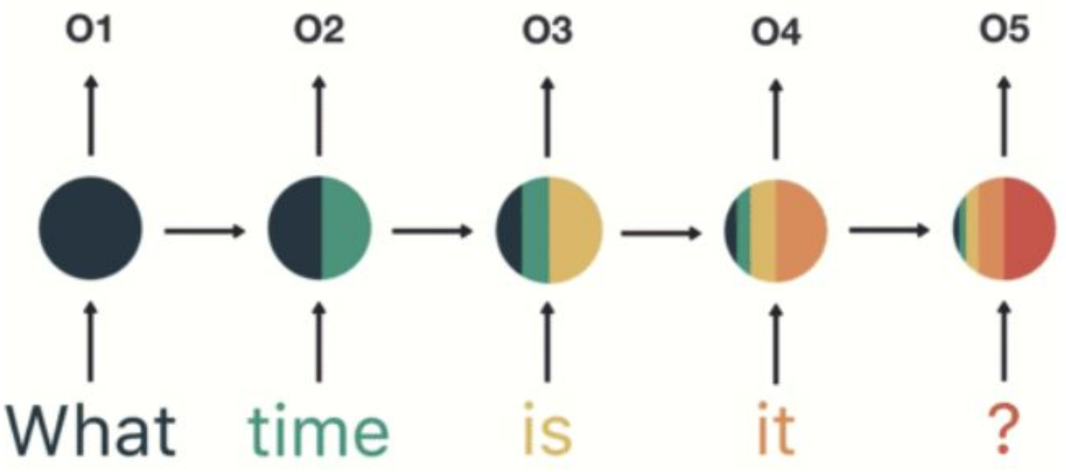
From RNN to Transformers



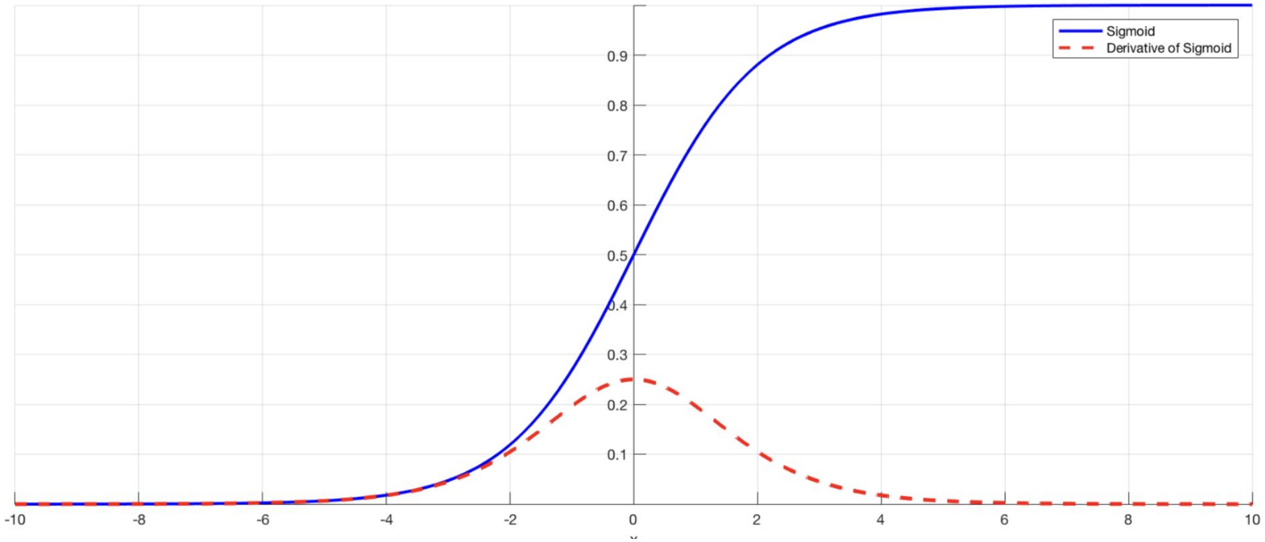


From RNN to Transformers





Problems with RNNs

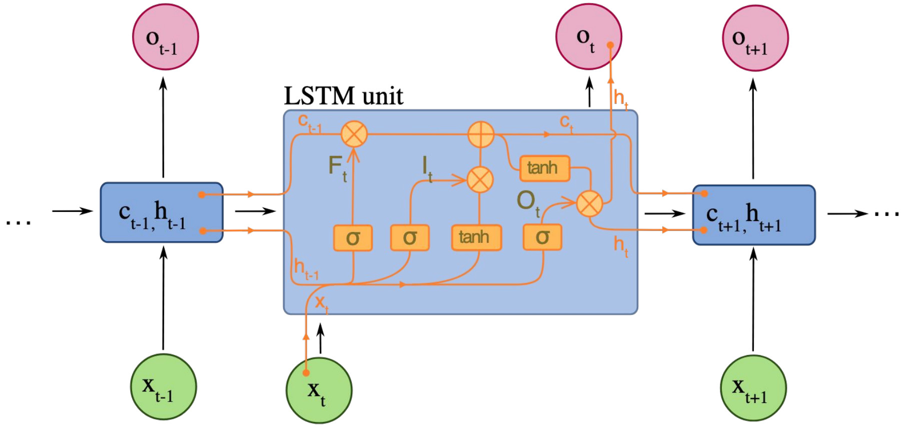


- **Vanishing Gradient**
- **Problem with long sequences**

LSTM (1997) / GRU (2014)

Input Gate / Output Gate / Forget Gate

- Part of the memory to drop
- New information to add to the memory
- Define the hidden state (for next step)



Limits of recurrent models

Difficult to parallelize on GPU

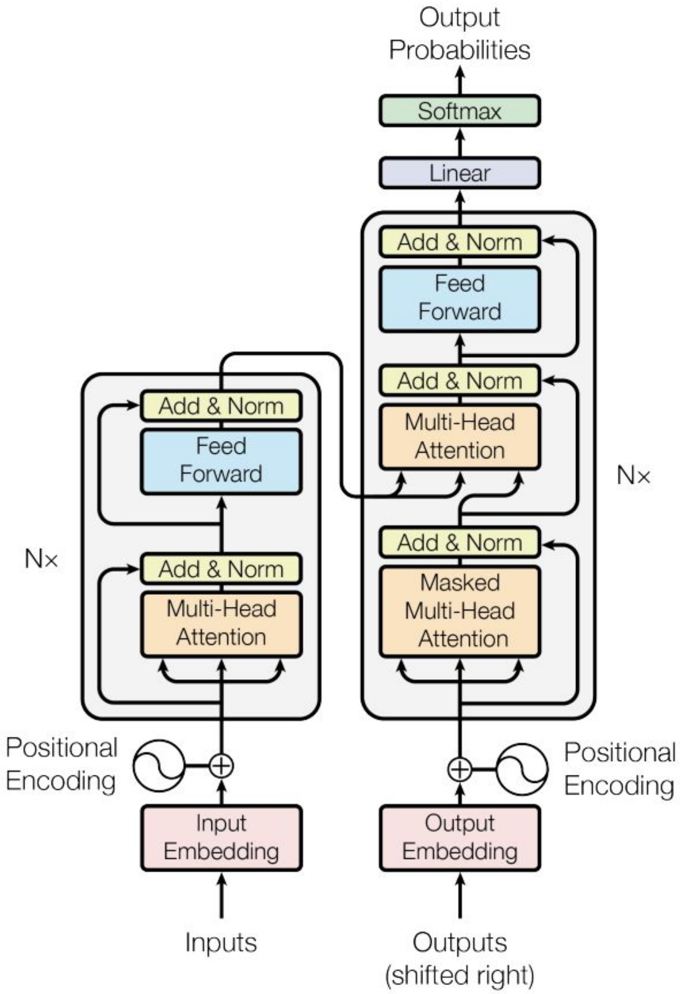
Easy overfitting

III

Transformers

Attention is all you need !

Global architecture



Encoder-Decoder architecture

Training example

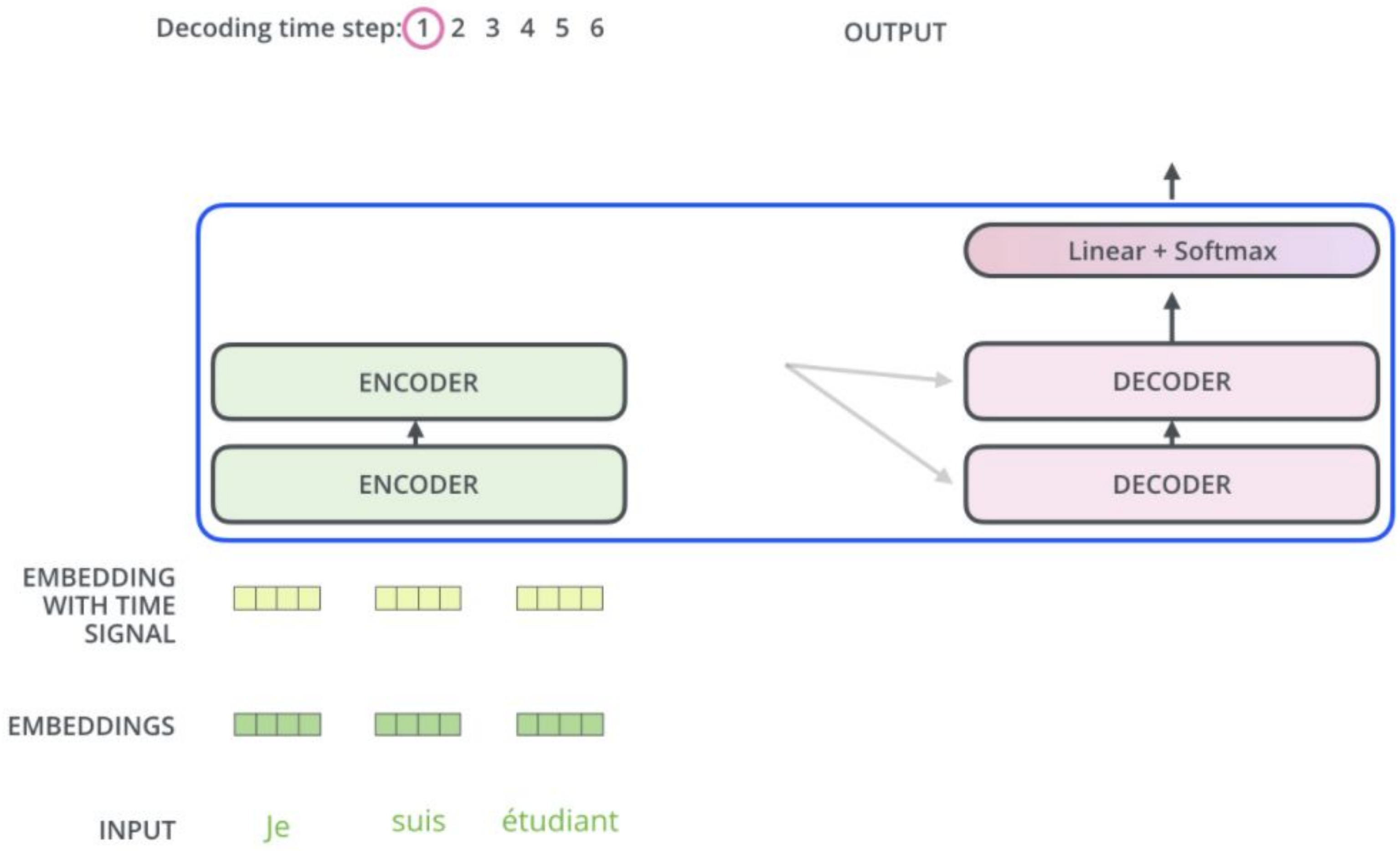


Image : https://jalammar.github.io/illustrated-transformer/

Training example

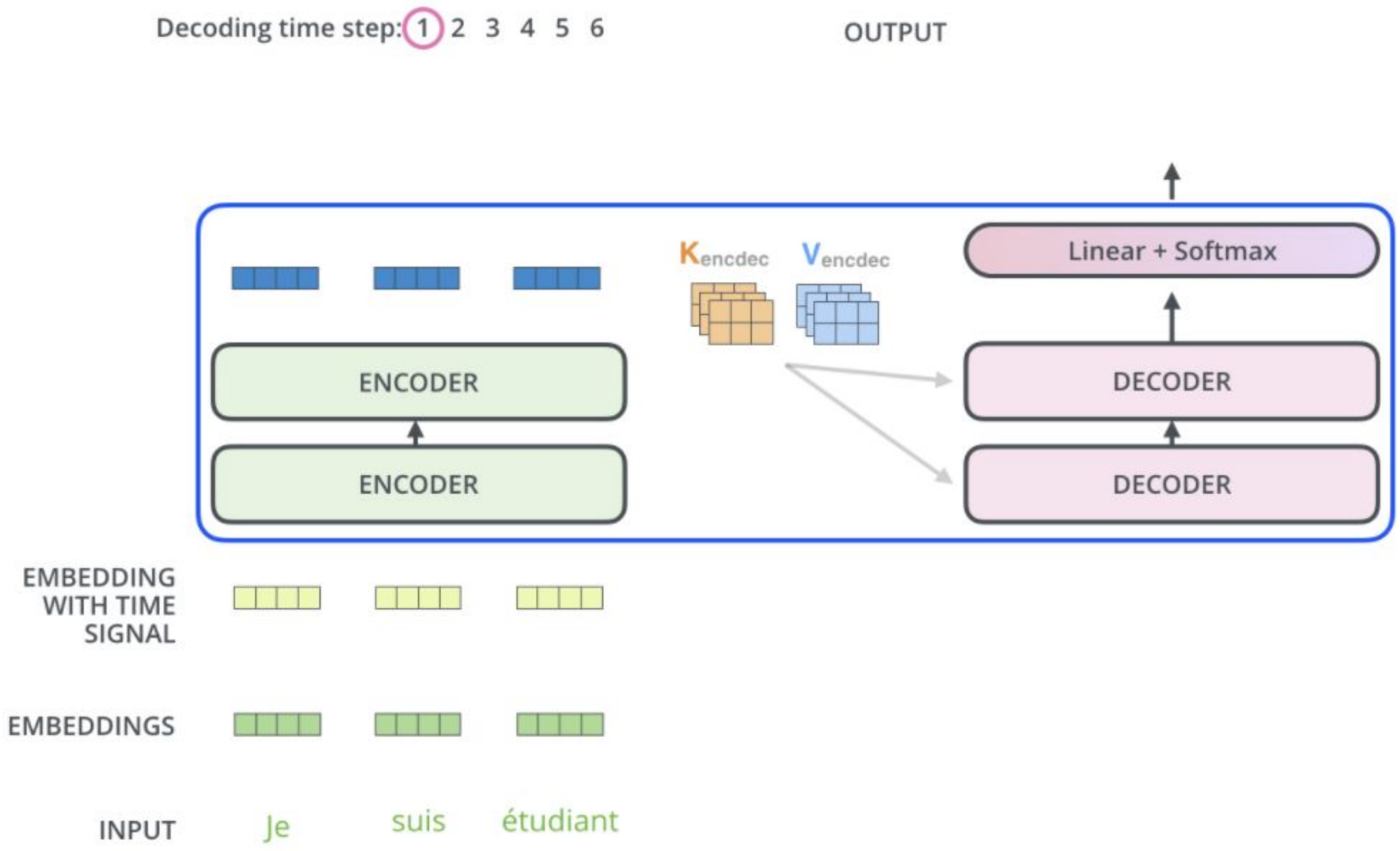


Image : https://jalammar.github.io/illustrated-transformer/

Training example

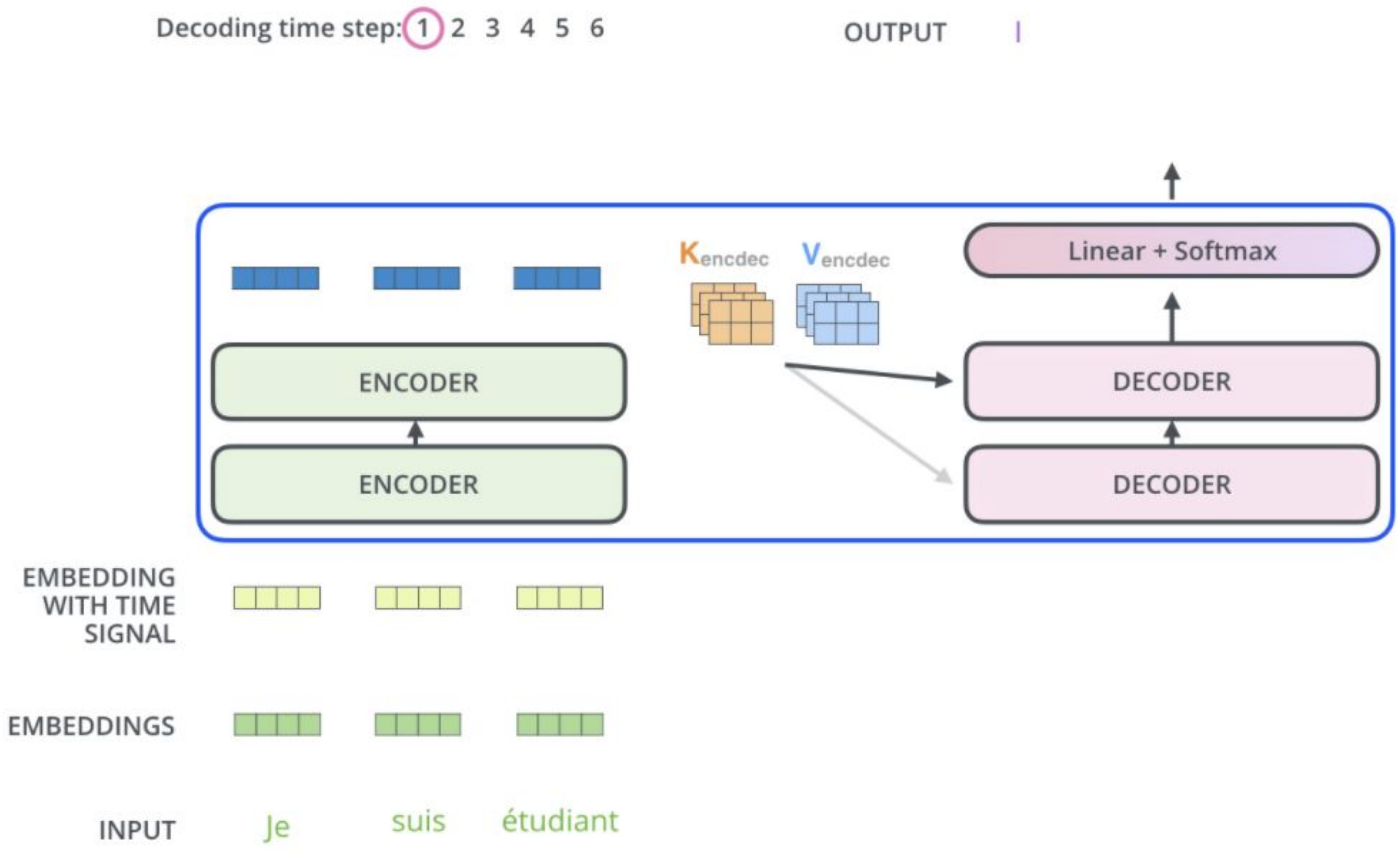


Image : https://jalammar.github.io/illustrated-transformer/

Training example

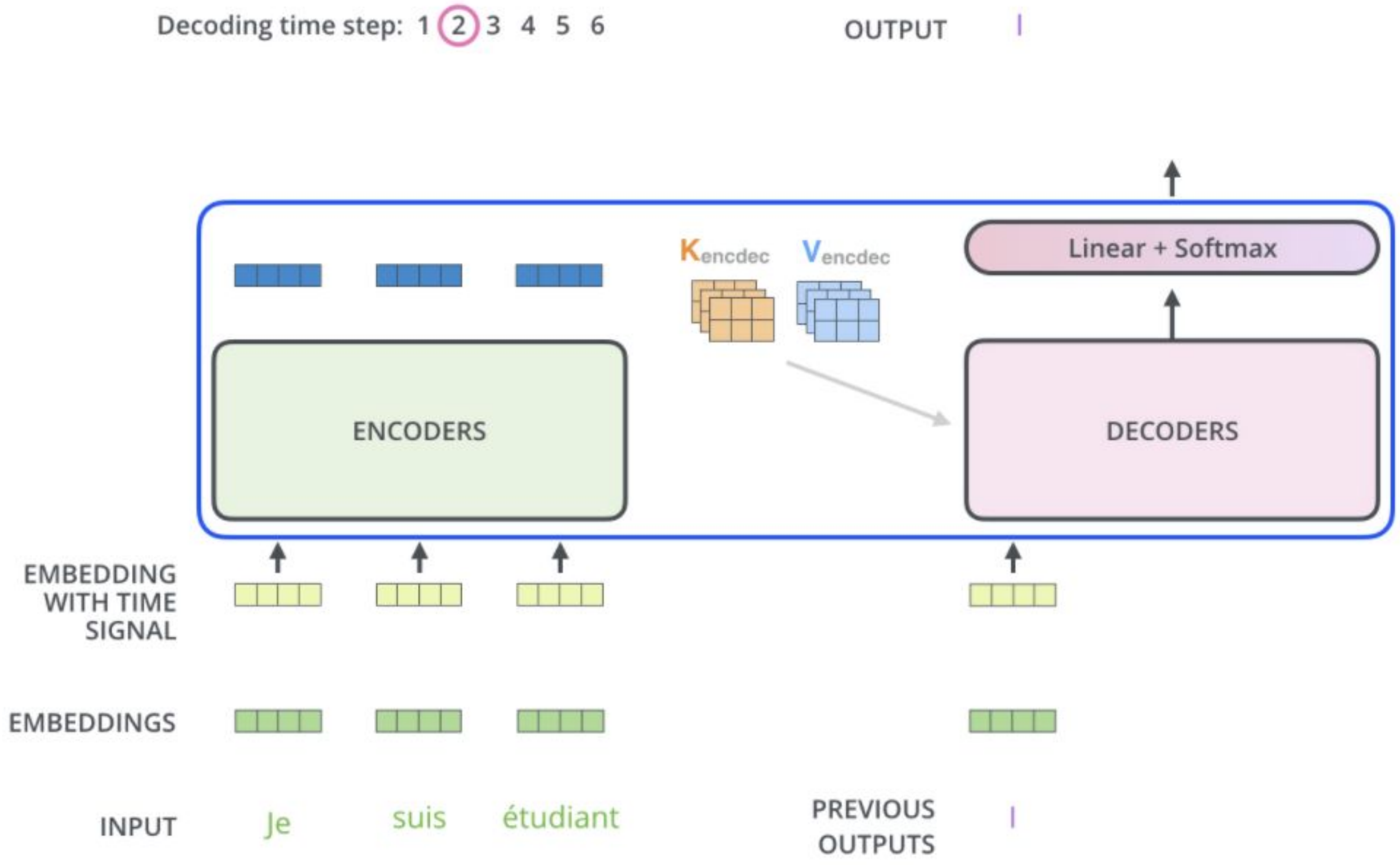


Image : https://jalammar.github.io/illustrated-transformer/

Training example

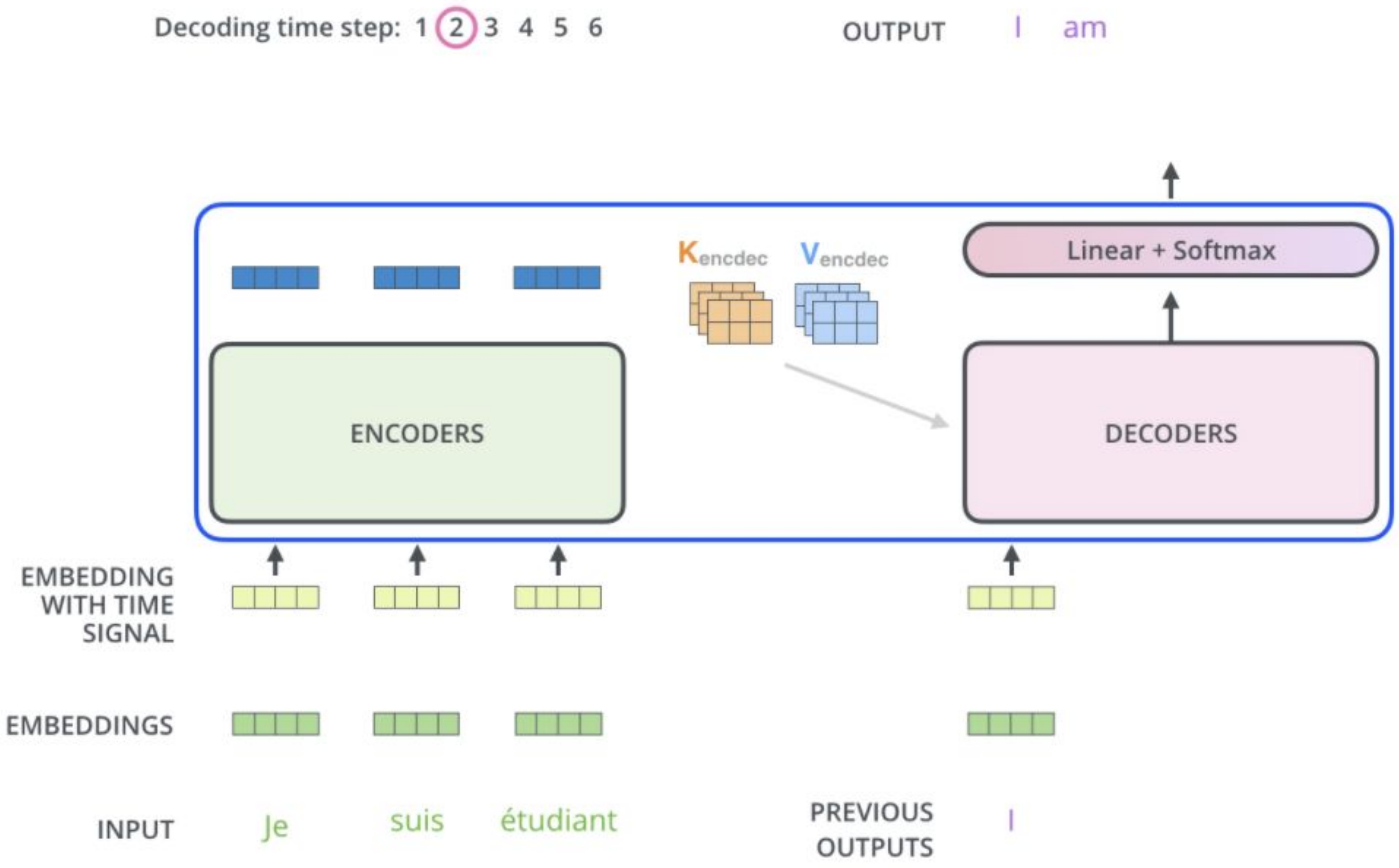


Image : https://jalammar.github.io/illustrated-transformer/

Training example

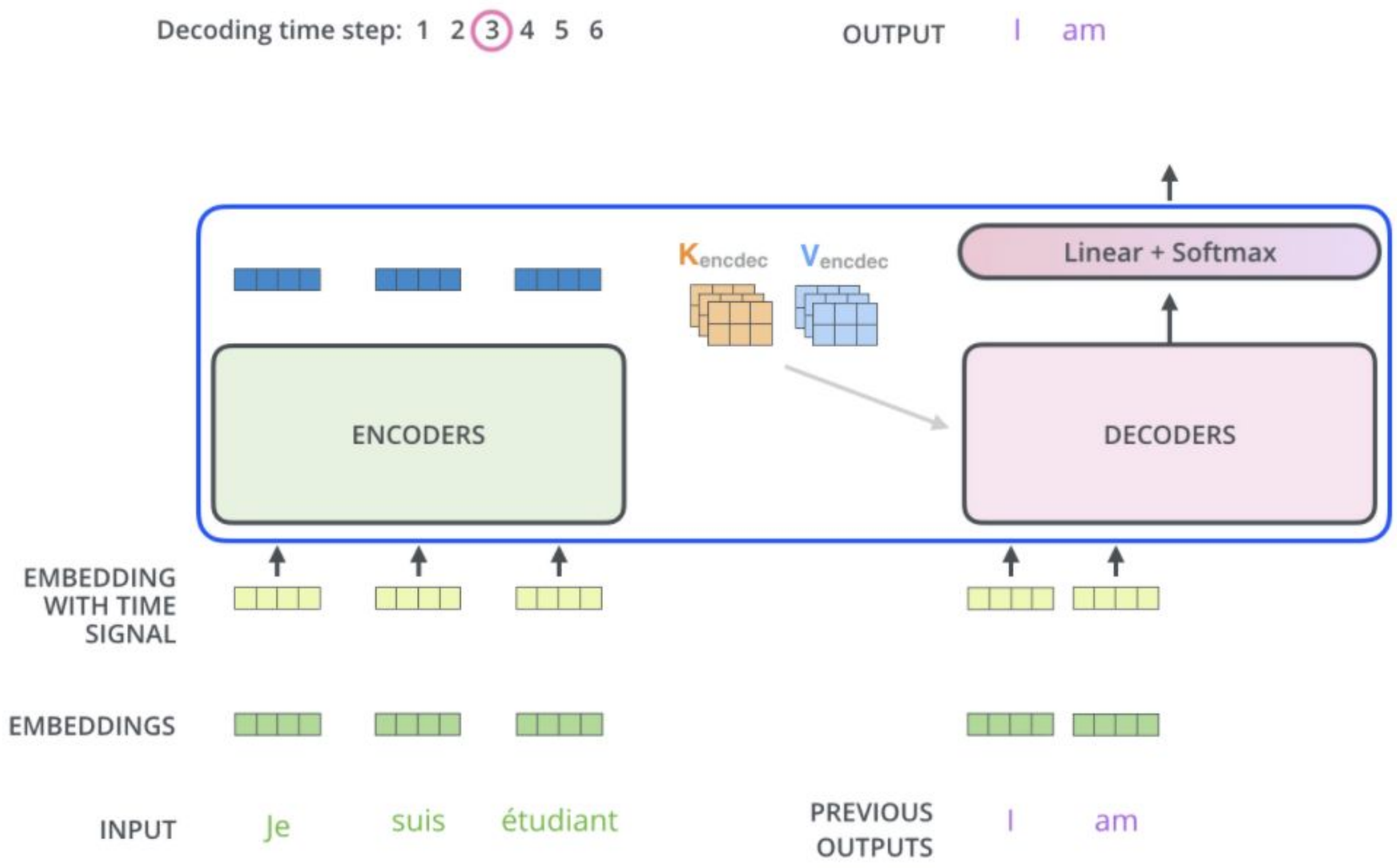


Image : https://jalammar.github.io/illustrated-transformer/

Training example

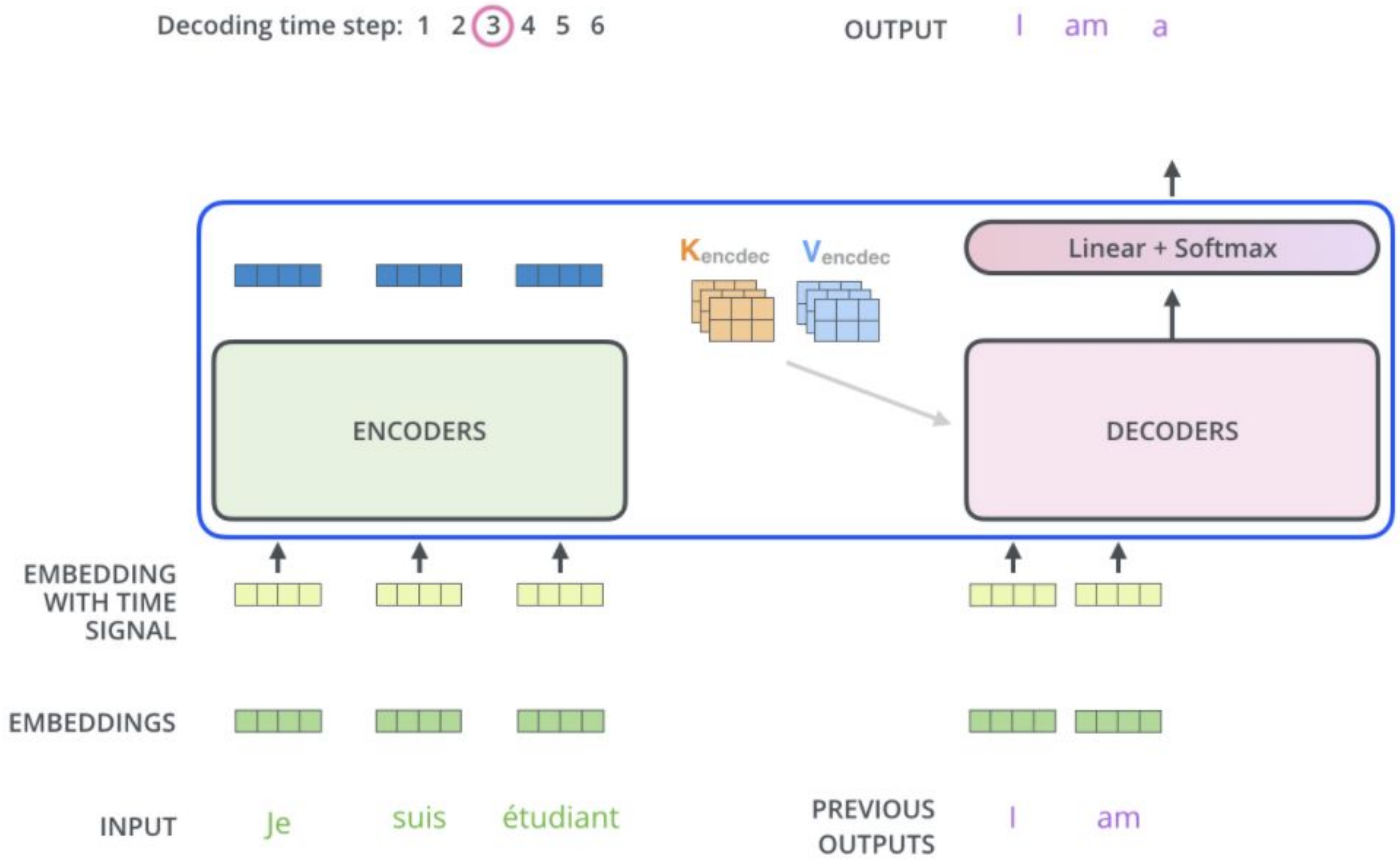


Image : https://jalammar.github.io/illustrated-transformer/

Training example

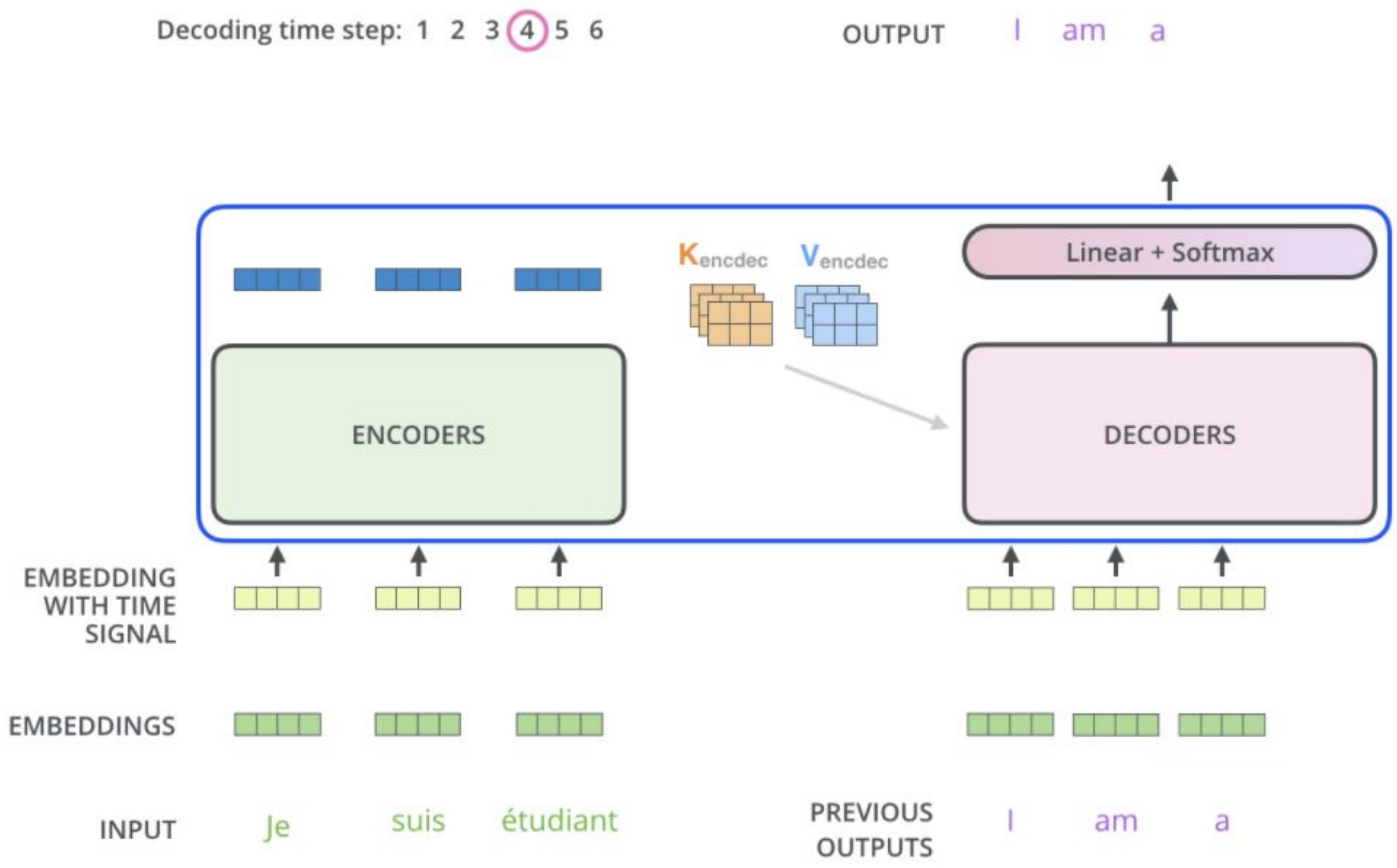


Image : https://jalammar.github.io/illustrated-transformer/

Training example

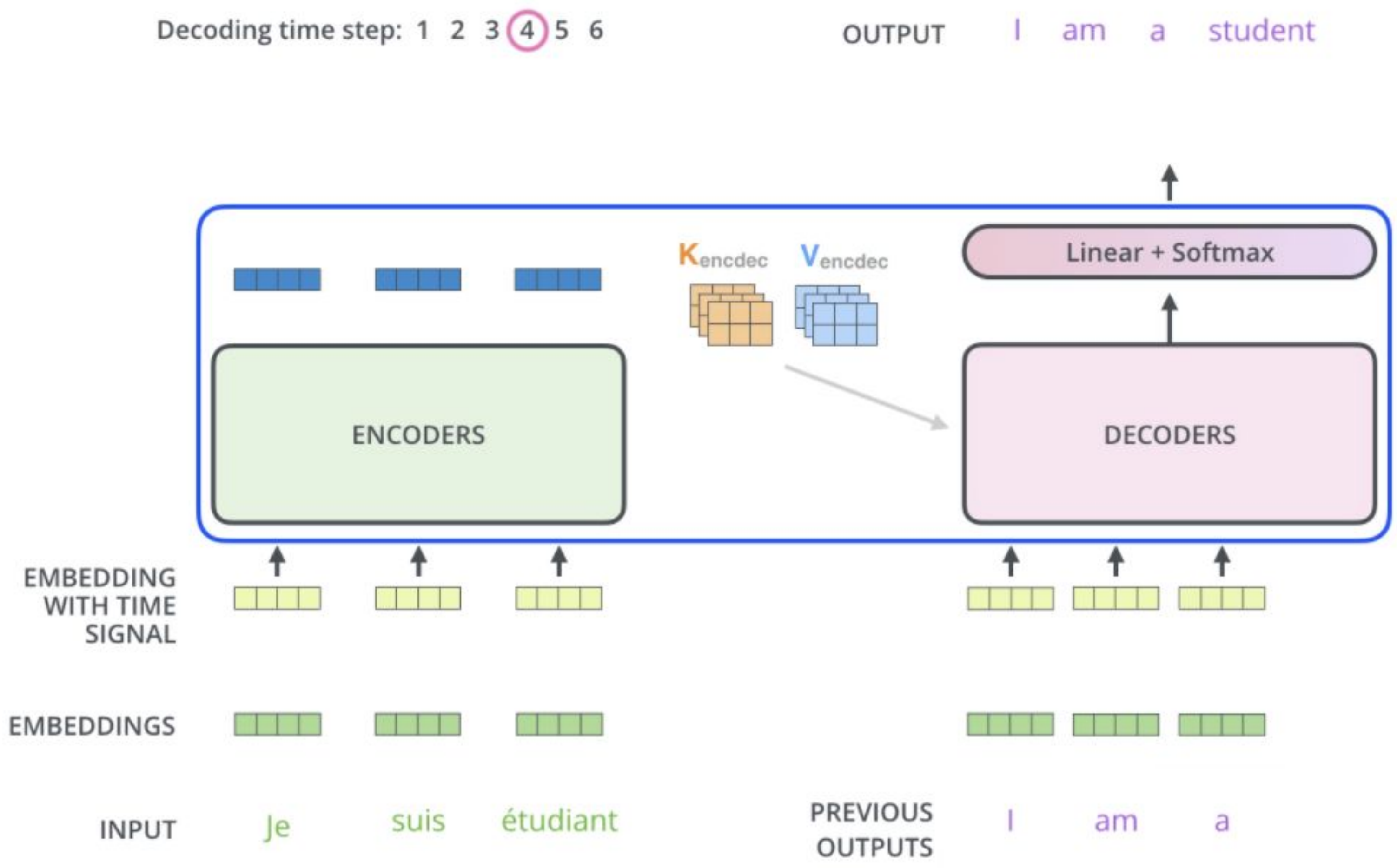


Image : https://jalammar.github.io/illustrated-transformer/

Training example

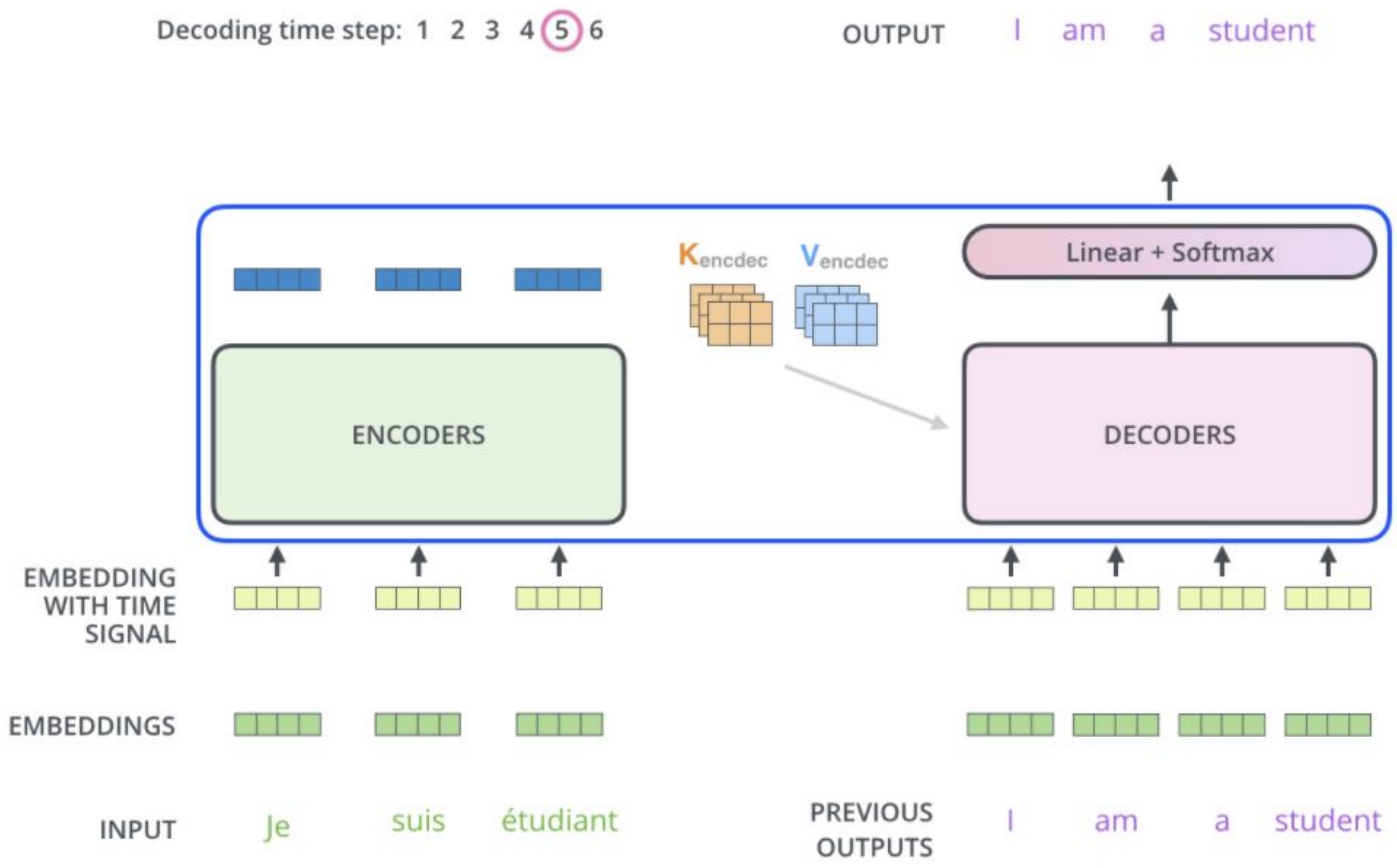


Image : https://jalammar.github.io/illustrated-transformer/

Training example

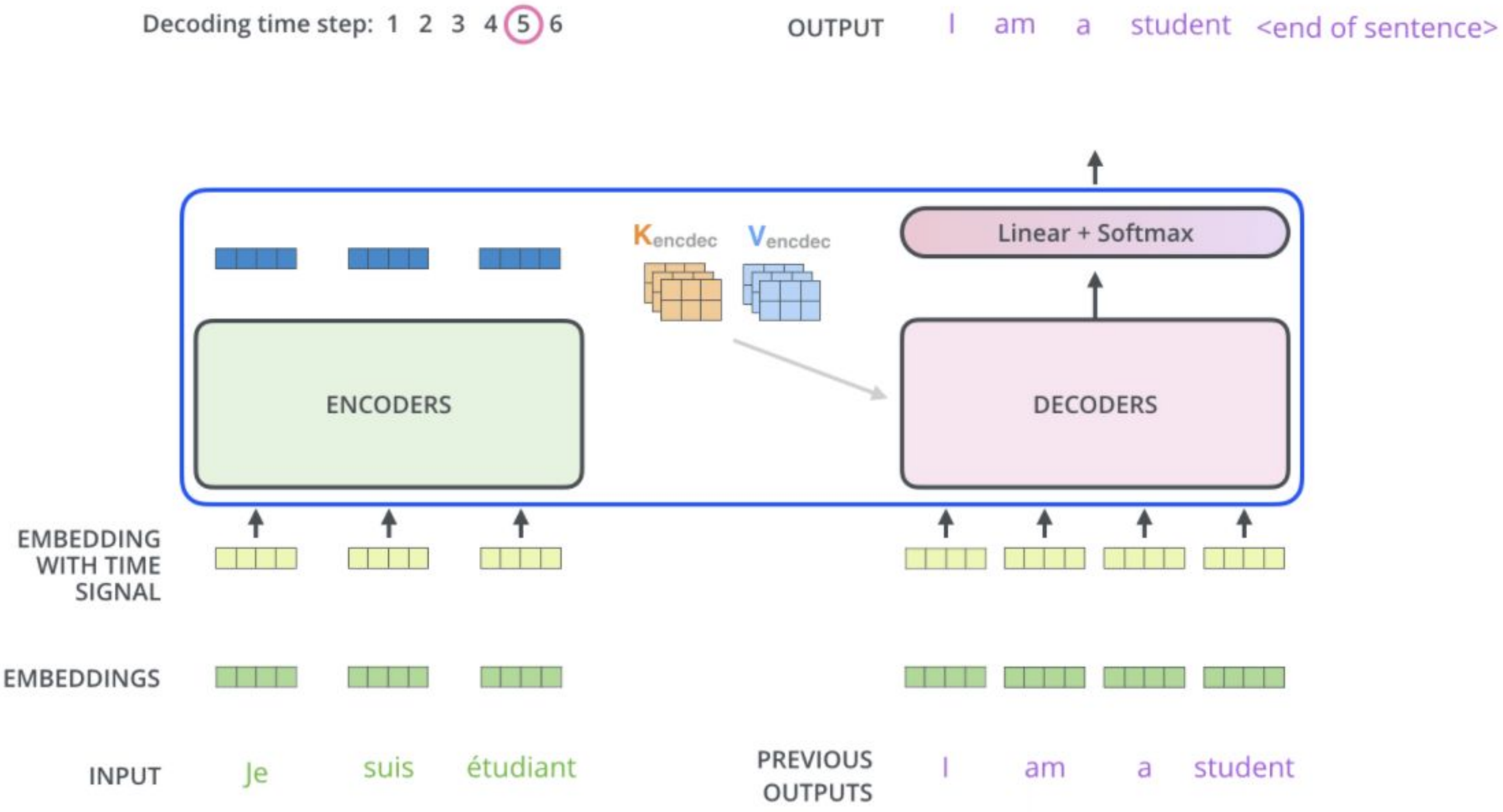
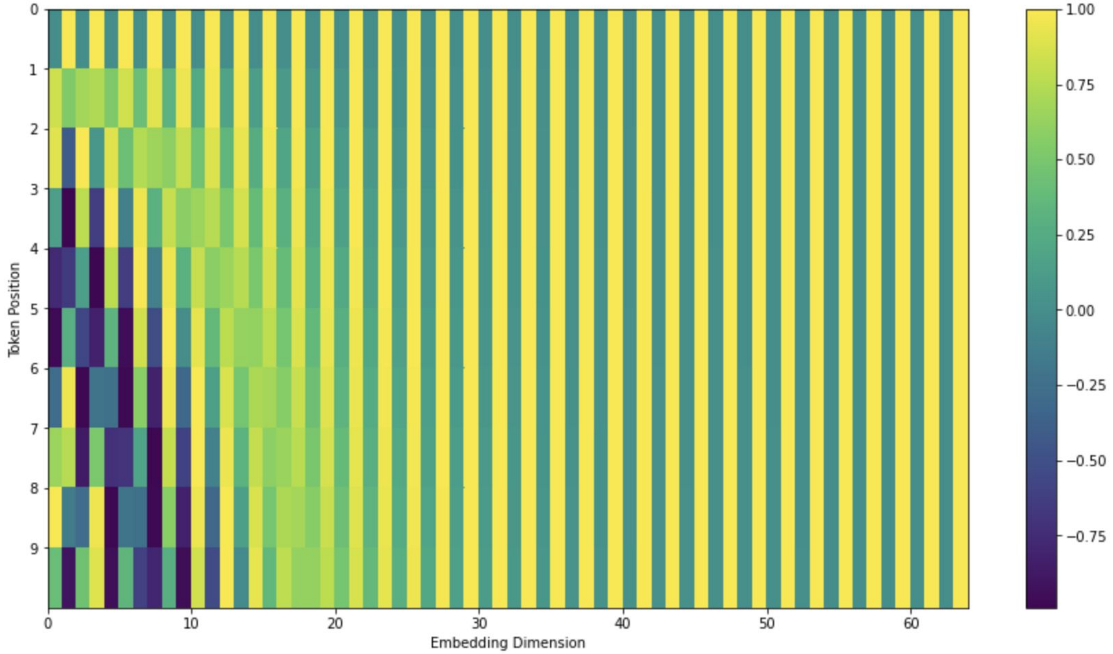


Image : https://jalammar.github.io/illustrated-transformer/

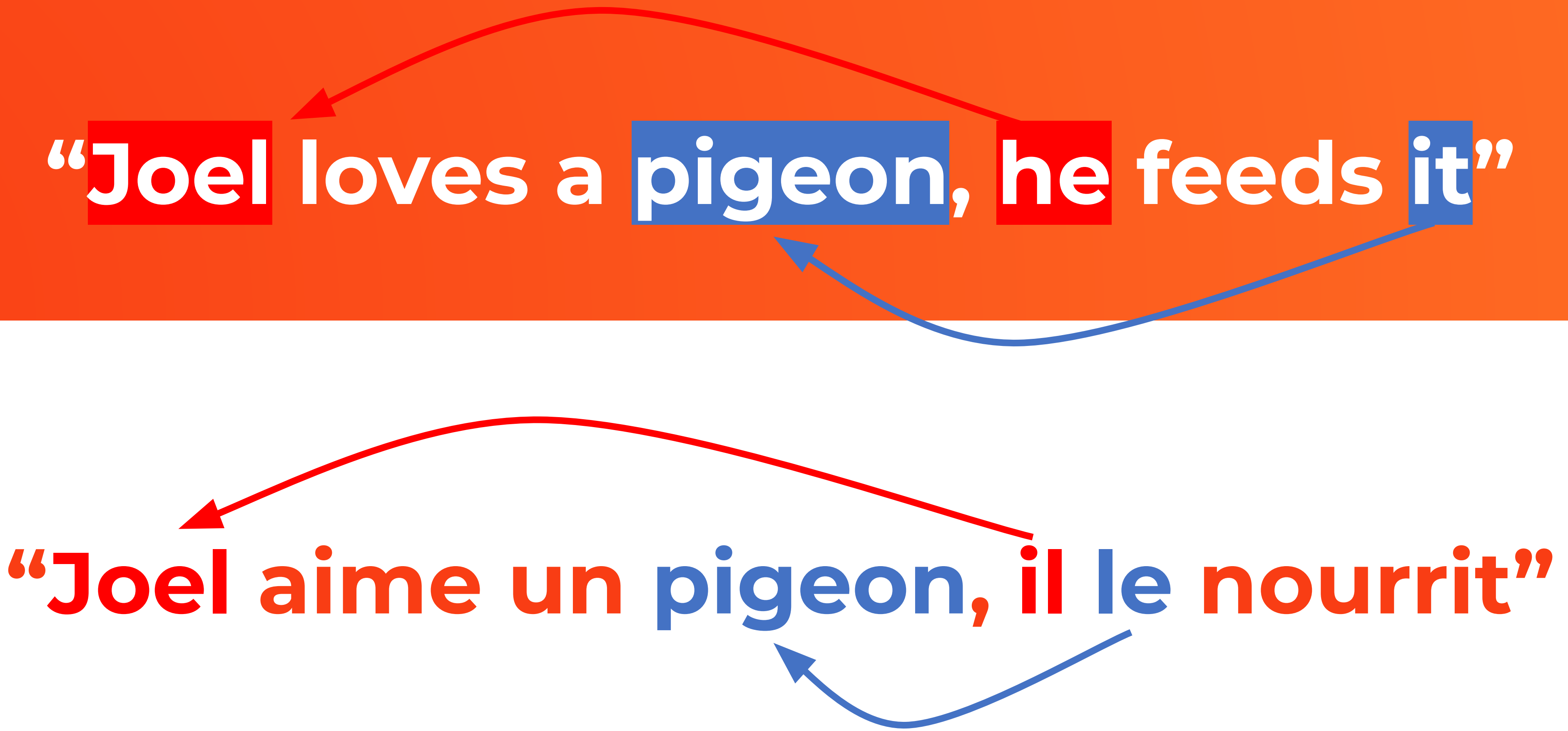
Positional encoding

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

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



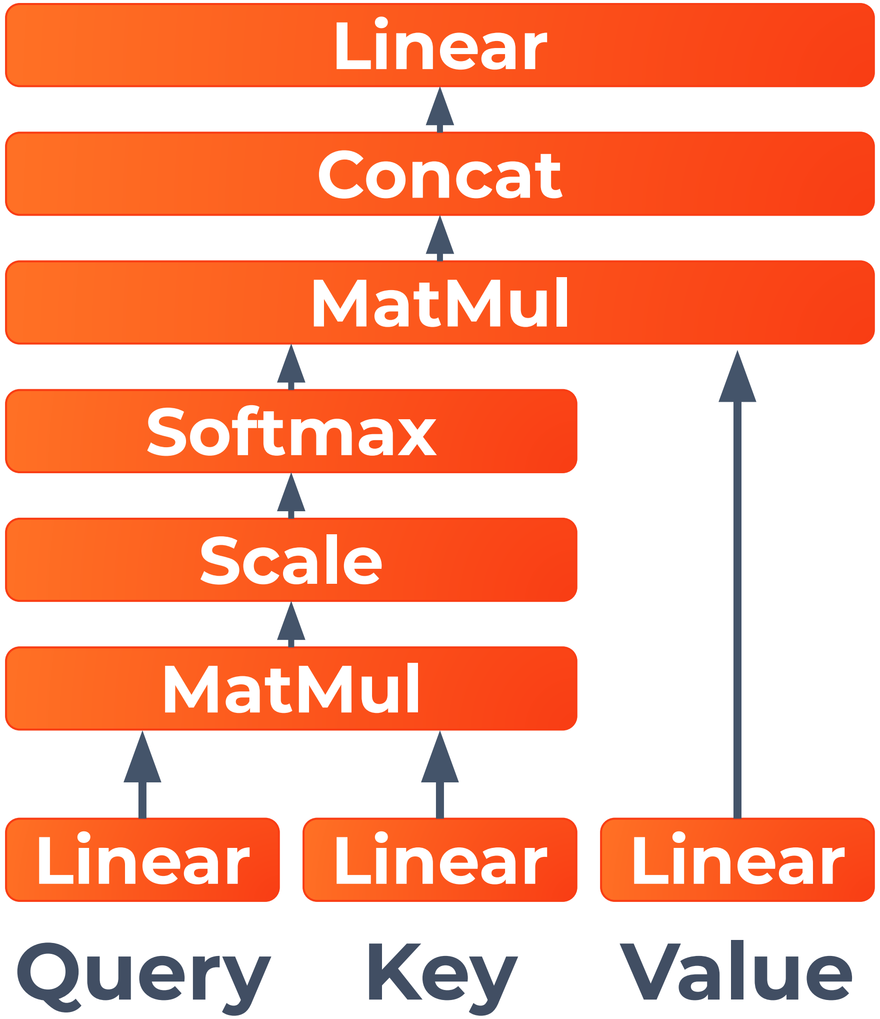
Attention mechanism

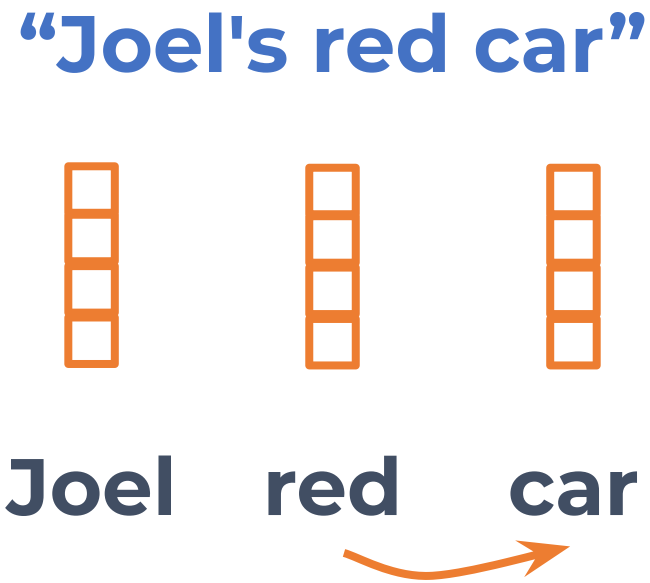


"Joel loves a pigeon, he feeds it"

"Joel aime un pigeon, il le nourrit"

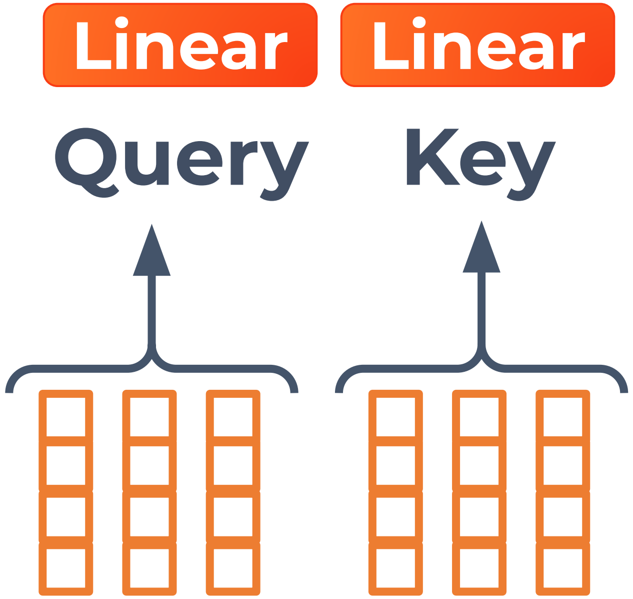
Self-Attention

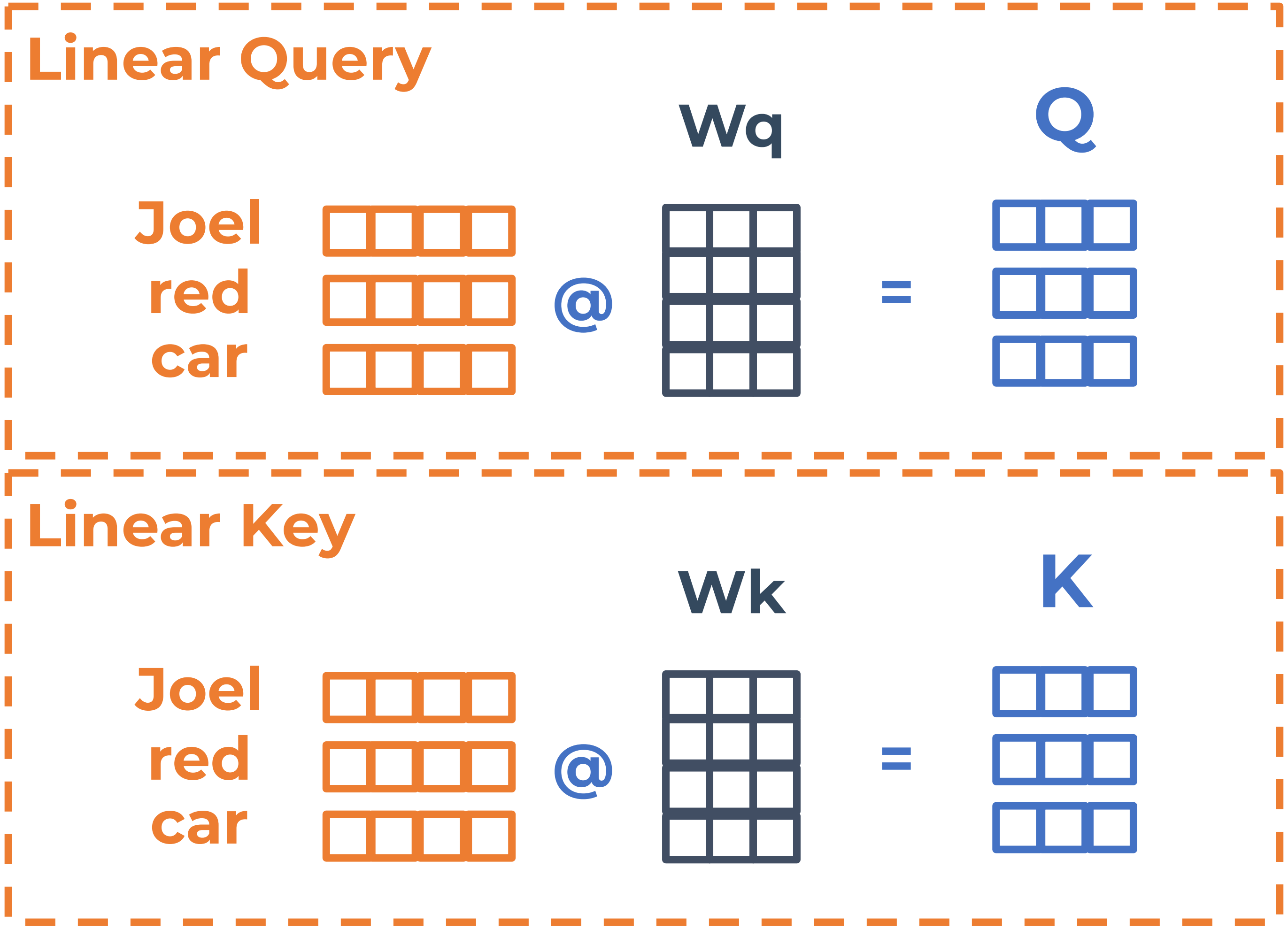




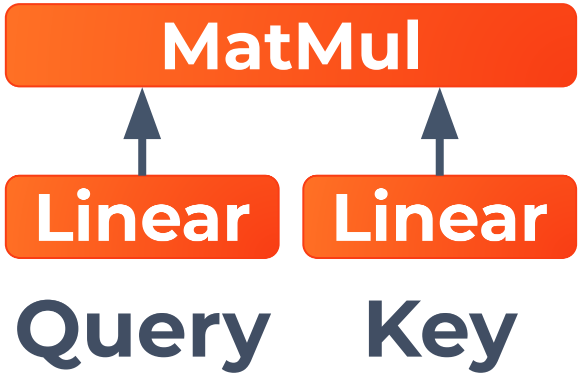
Query ~ Word we want know Attention
Key ~ All others Words
Value ~ Focused words

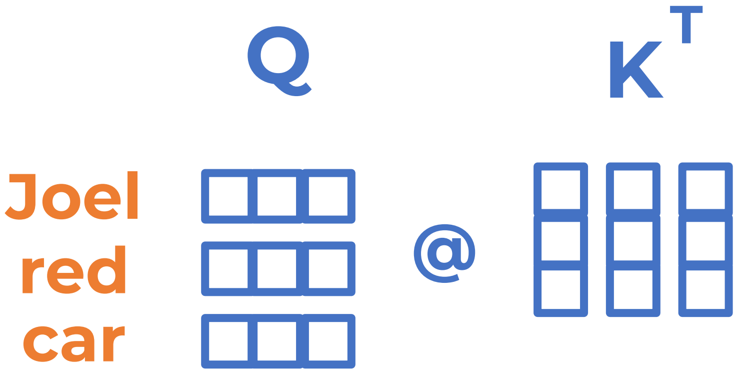
Self-Attention





Self-Attention

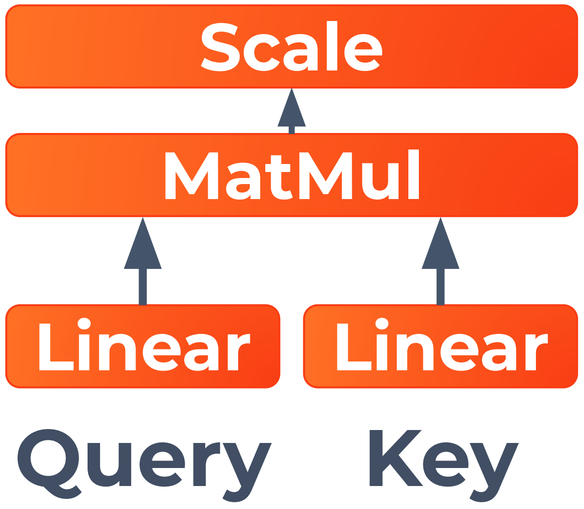




=

	Joel	red	car
Joel	2.8	0.6	0.2
red	0.3	2.1	1.6
car	0.2	1.4	2.6

Self-Attention



Scale

MatMul

Linear

Linear

Query

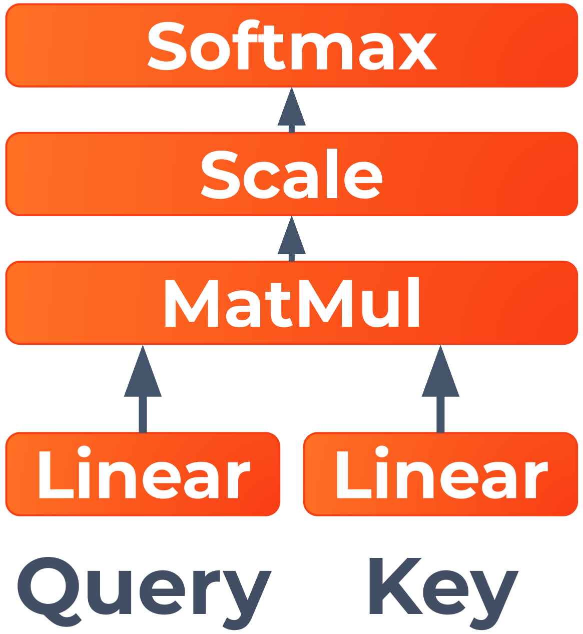
Key

$$\begin{array}{c|c|c|c|} & \text{Joel} & \text{red} & \text{car} \\ \hline \text{Joel} & 2.8 & 0.6 & 0.2 \\ \hline \text{red} & 0.3 & 2.1 & 1.6 \\ \hline \text{car} & 0.2 & 1.4 & 2.6 \\ \hline \end{array} \;/\; \sqrt{(dk)}$$

dk = the square root of the dimension of the key vectors

➡ More stable gradients !

Self-Attention



Softmax

Scale

MatMul

Linear

Linear

Query

Key

$$\text{Softmax}\left(\begin{array}{c|ccc} & \text{Joel} & \text{red} & \text{car} \\ \hline \text{Joel} & 1.6 & 0.3 & 0.1 \\ \text{red} & 0.2 & 1.2 & 0.9 \\ \text{car} & 0.1 & 0.8 & 1.5 \end{array}\right) = \begin{array}{ccc} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{array}$$

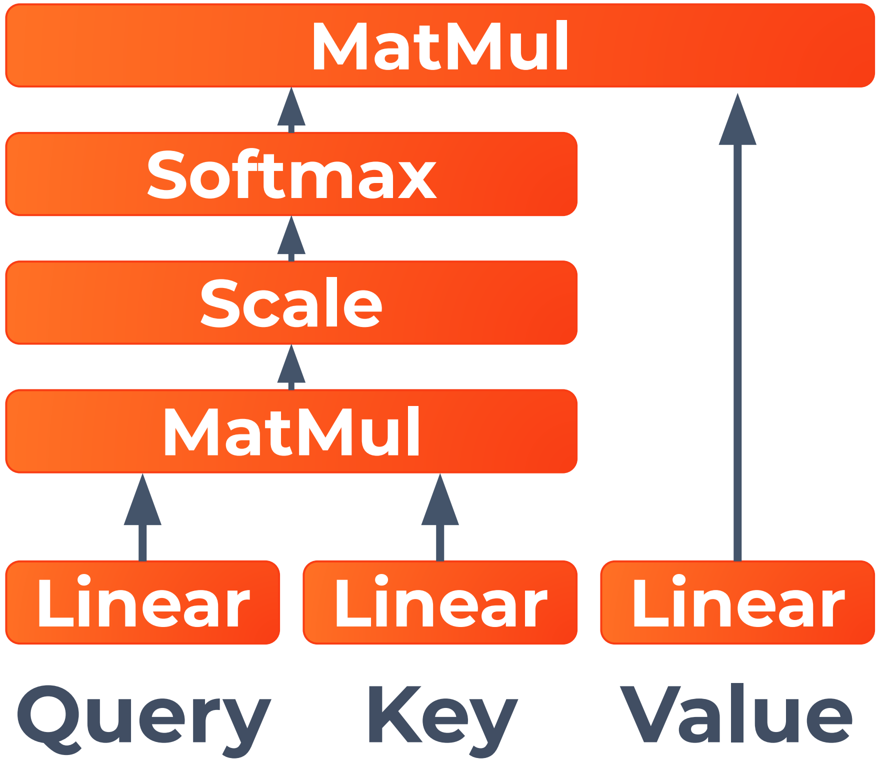
$0.7 + 0.2 + 0.1 = 1$

$0.2 + 0.5 + 0.3 = 1$

$0.1 + 0.3 + 0.6 = 1$

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K.$$

Self-Attention



MatMul

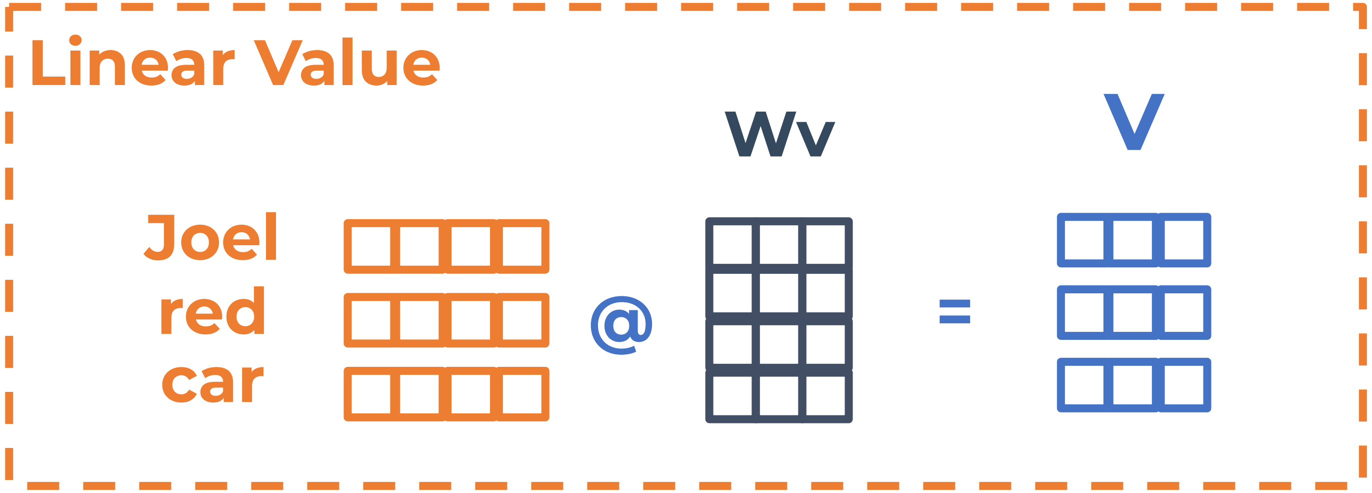
Softmax

Scale

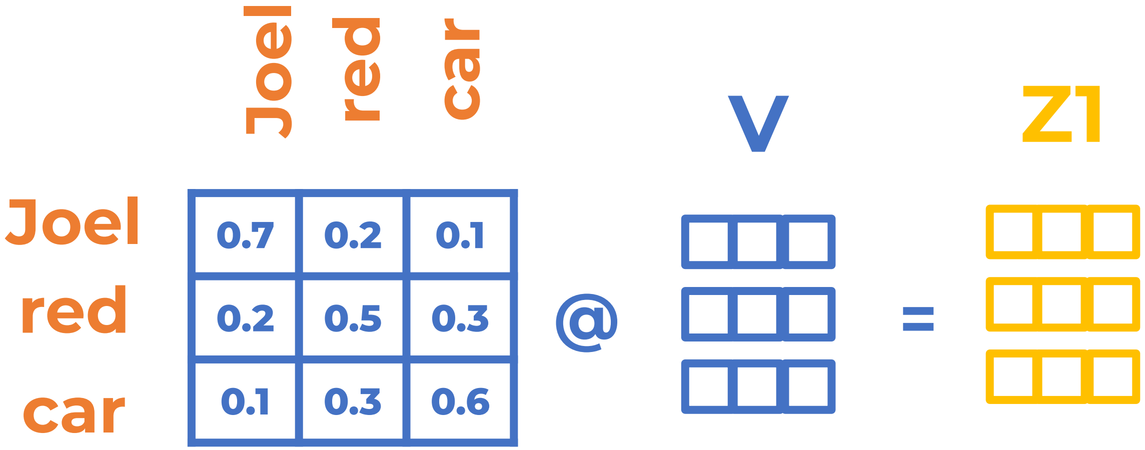
MatMul

Linear Linear Linear

Query Key Value

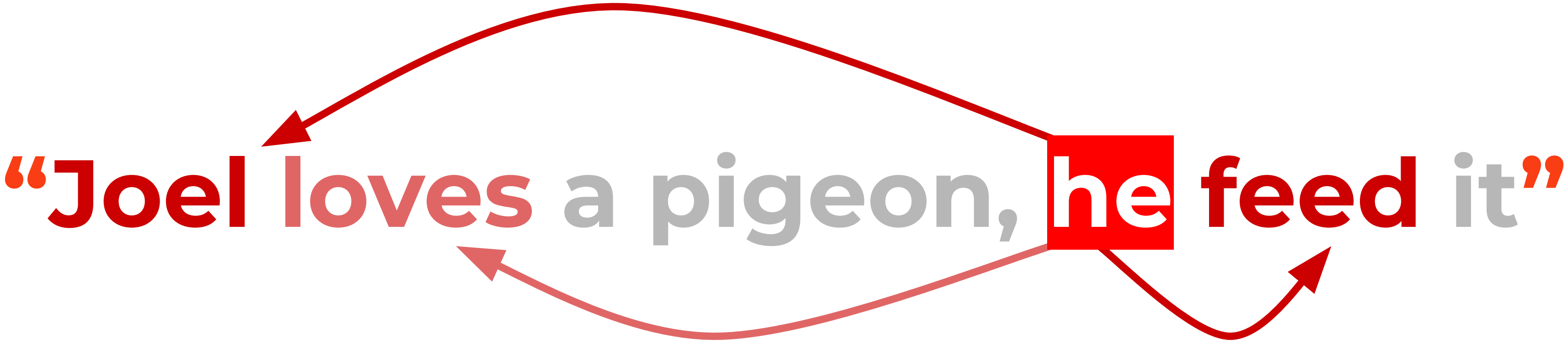


Linear Value



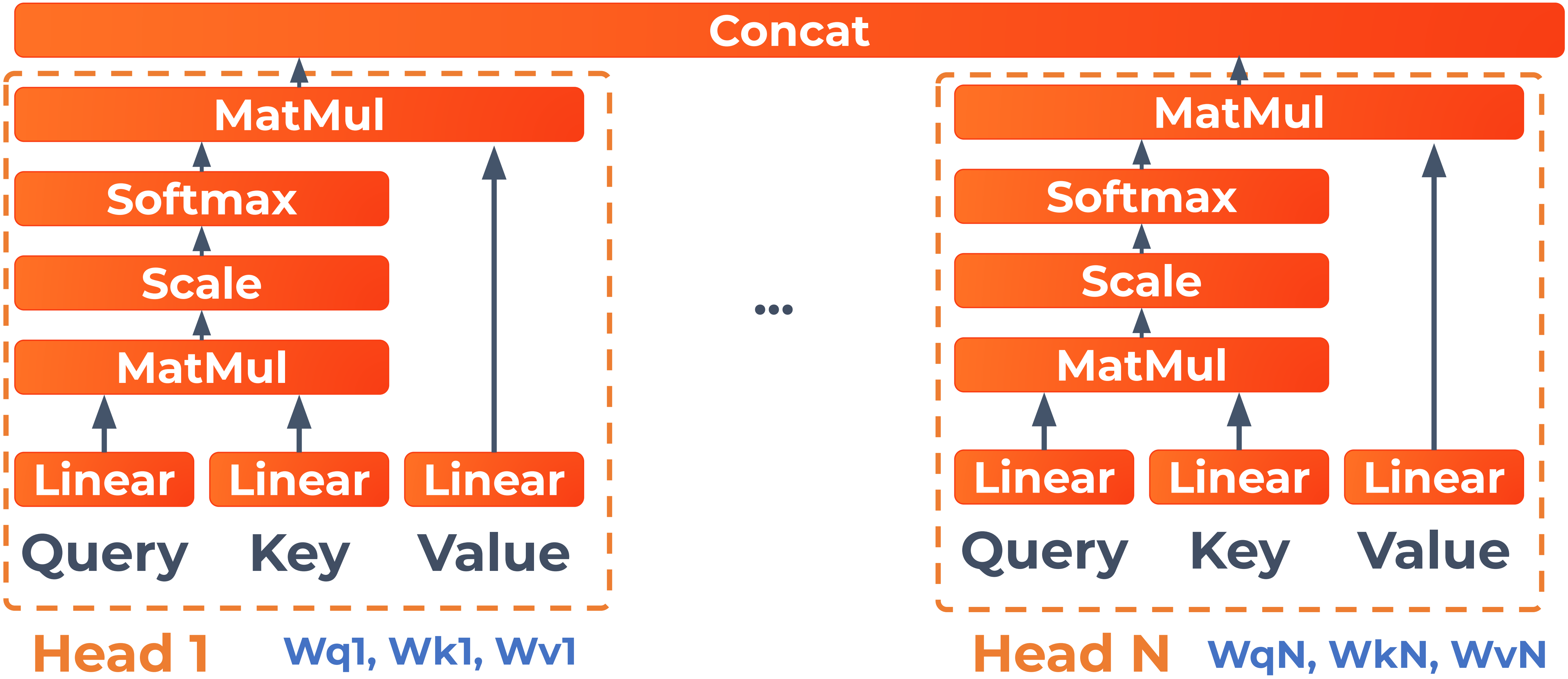
	Joel	red	car
Joel	0.7	0.2	0.1
red	0.2	0.5	0.3
car	0.1	0.3	0.6

Multi-head Attention

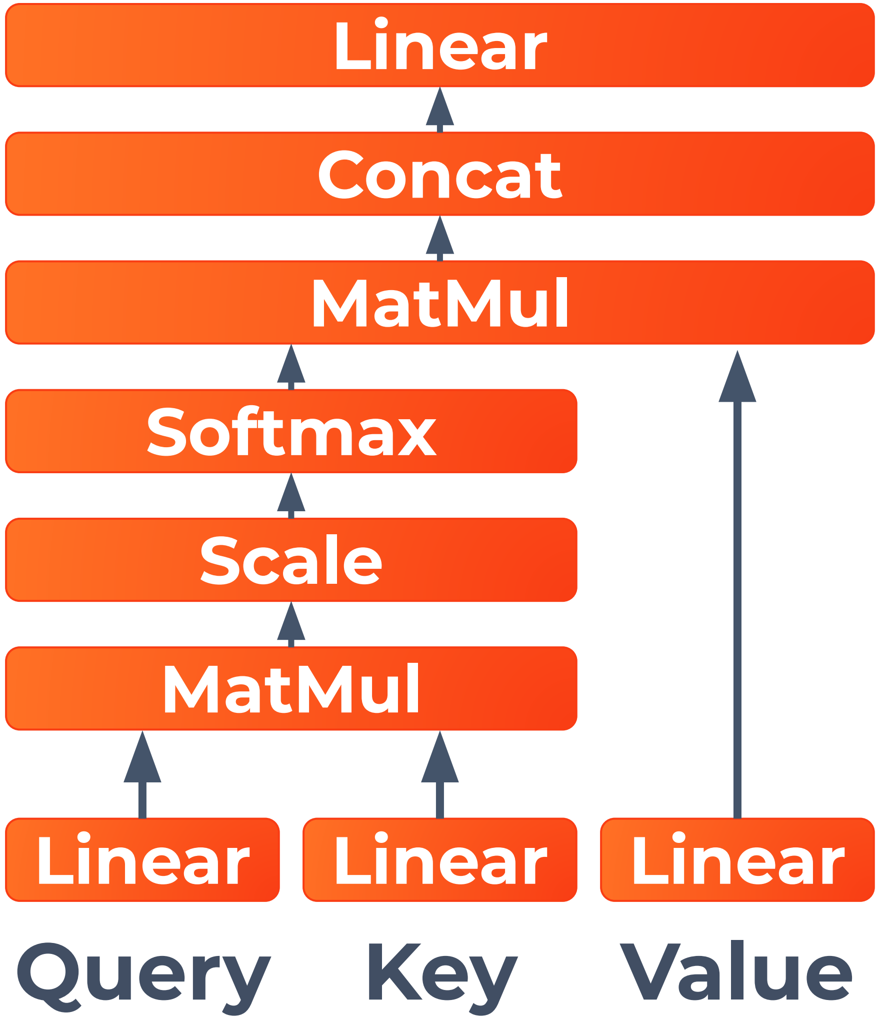


"Joel loves a pigeon, he feed it"

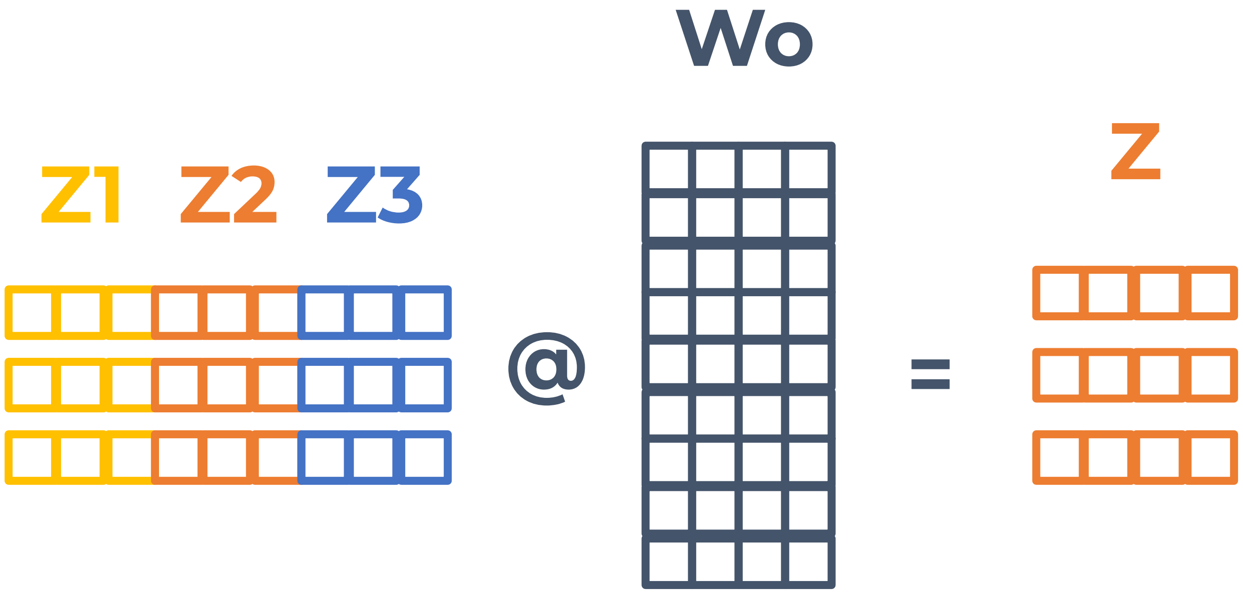
Multi-head Attention



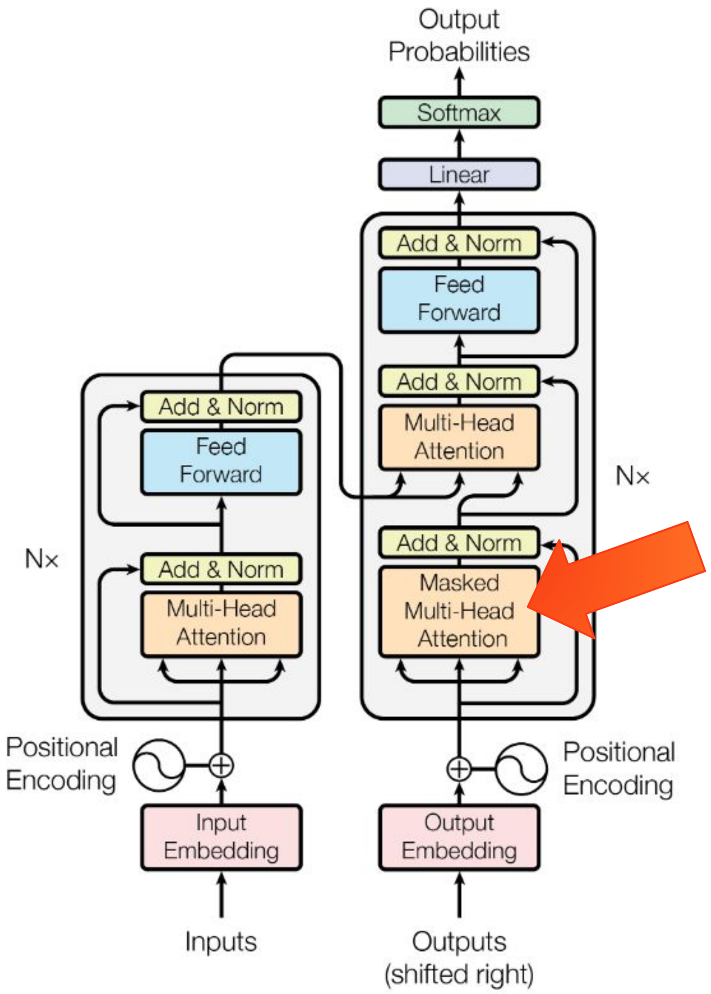
Multi-head Attention



For N=3 heads



Masked Multi-head Attention



Masking "future" values to avoid leaks

	Joel	red	car
Joel	2.8	-inf	-inf
red	0.3	2.1	-inf
car	0.2	1.4	2.6

	Joel	red	car
Joel	1	0	0
red	0.3	0.7	0
car	0.1	0.3	0.6

Joel red car

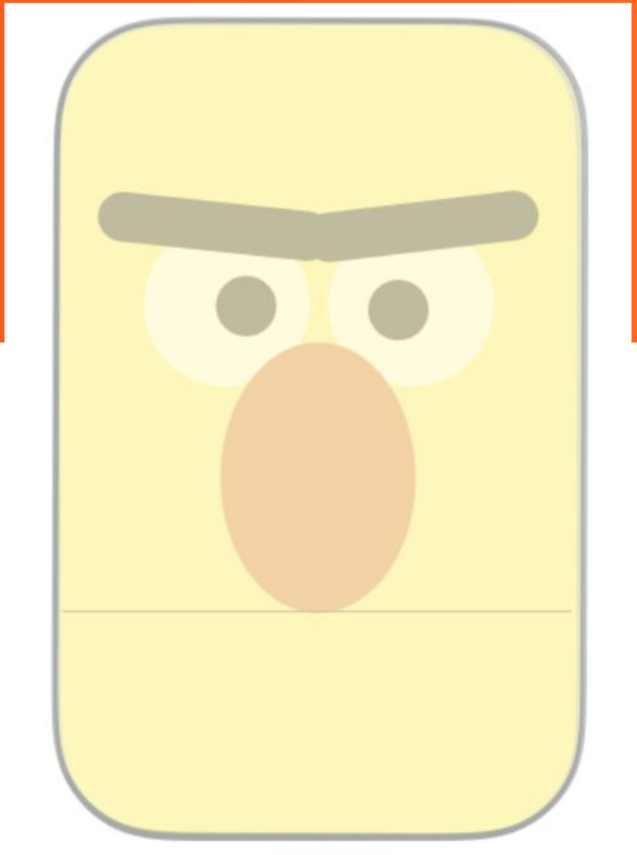
Future !

Joel red car

Hate Speech detection

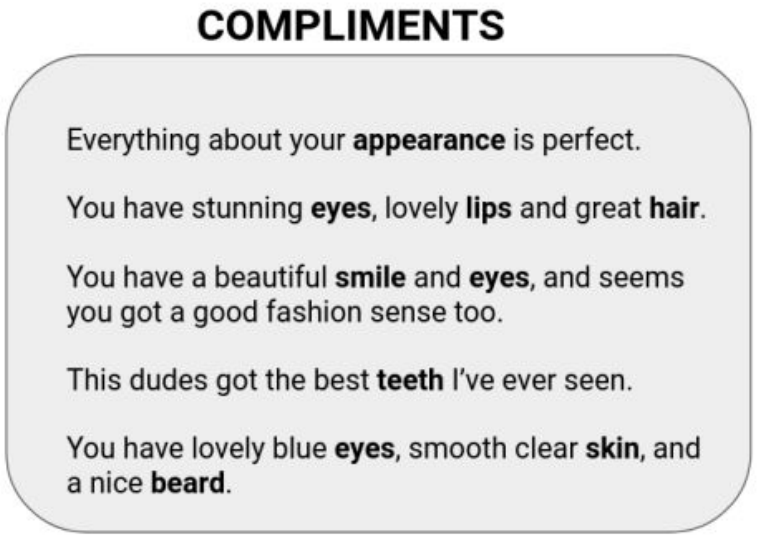
- Reddit Dataset: Jibes & Delights (2021)

- HateBERT



$\text{BERT}_{\text{LARGE}}$

Dataset: Jibes & Delights (2021)



COMPLIMENTS

Everything about your **appearance** is perfect.

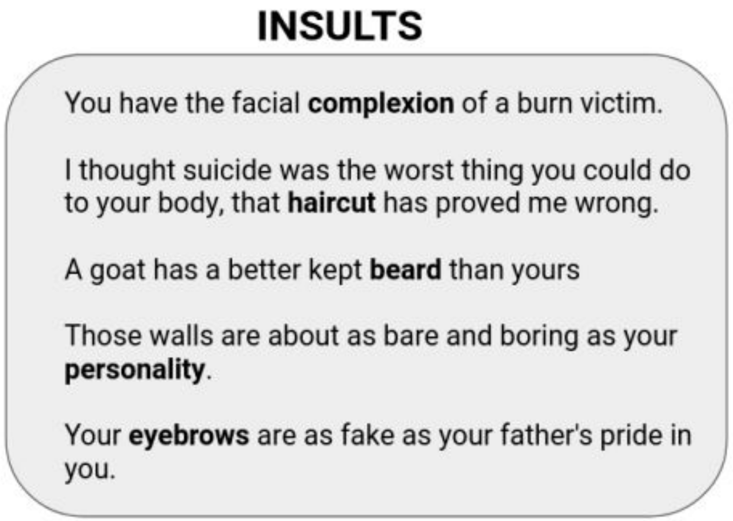
You have stunning **eyes**, lovely **lips** and great **hair**.

You have a beautiful **smile** and **eyes**, and seems you got a good fashion sense too.

This dudes got the best **teeth** I've ever seen.

You have lovely blue **eyes**, smooth clear **skin**, and a nice **beard**.

ToastMe / FreeCompliments



INSULTS

You have the facial **complexion** of a burn victim.

I thought suicide was the worst thing you could do to your body, that **haircut** has proved me wrong.

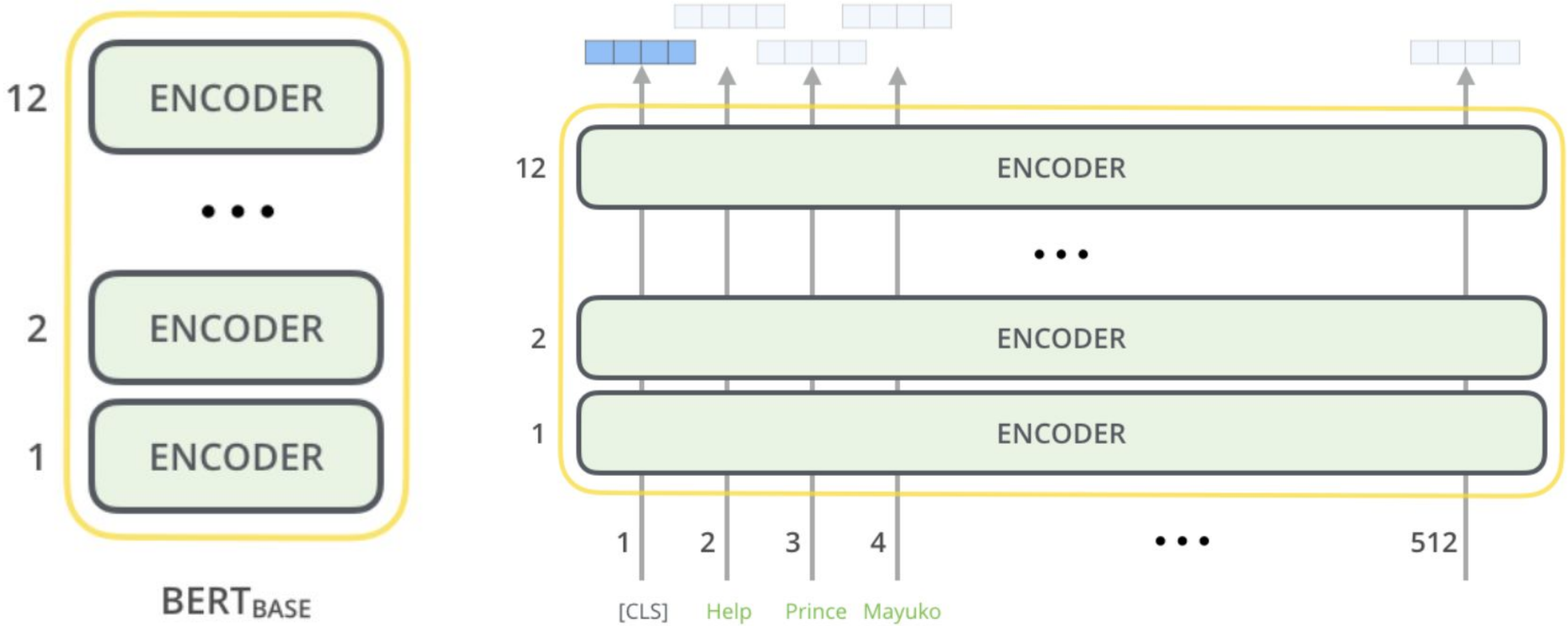
A goat has a better kept **beard** than yours

Those walls are about as bare and boring as your **personality**.

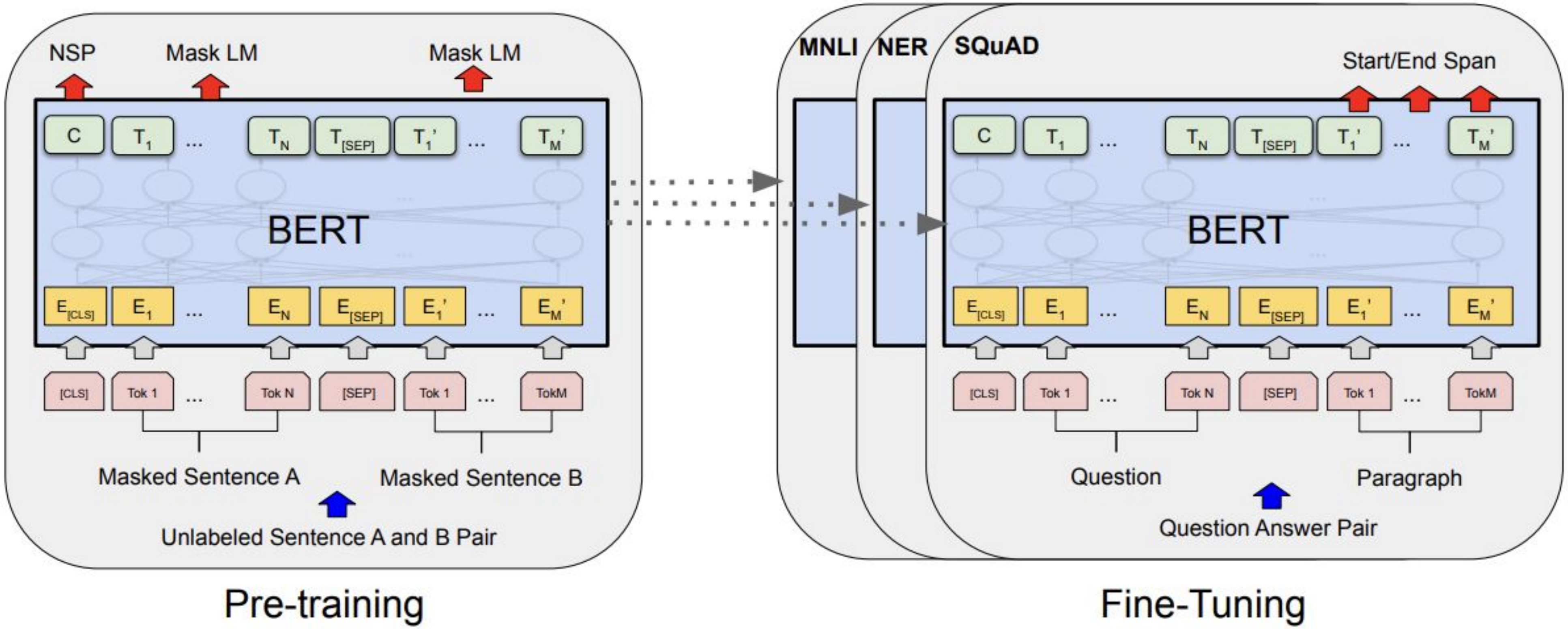
Your **eyebrows** are as fake as your father's pride in you.

RoastMe

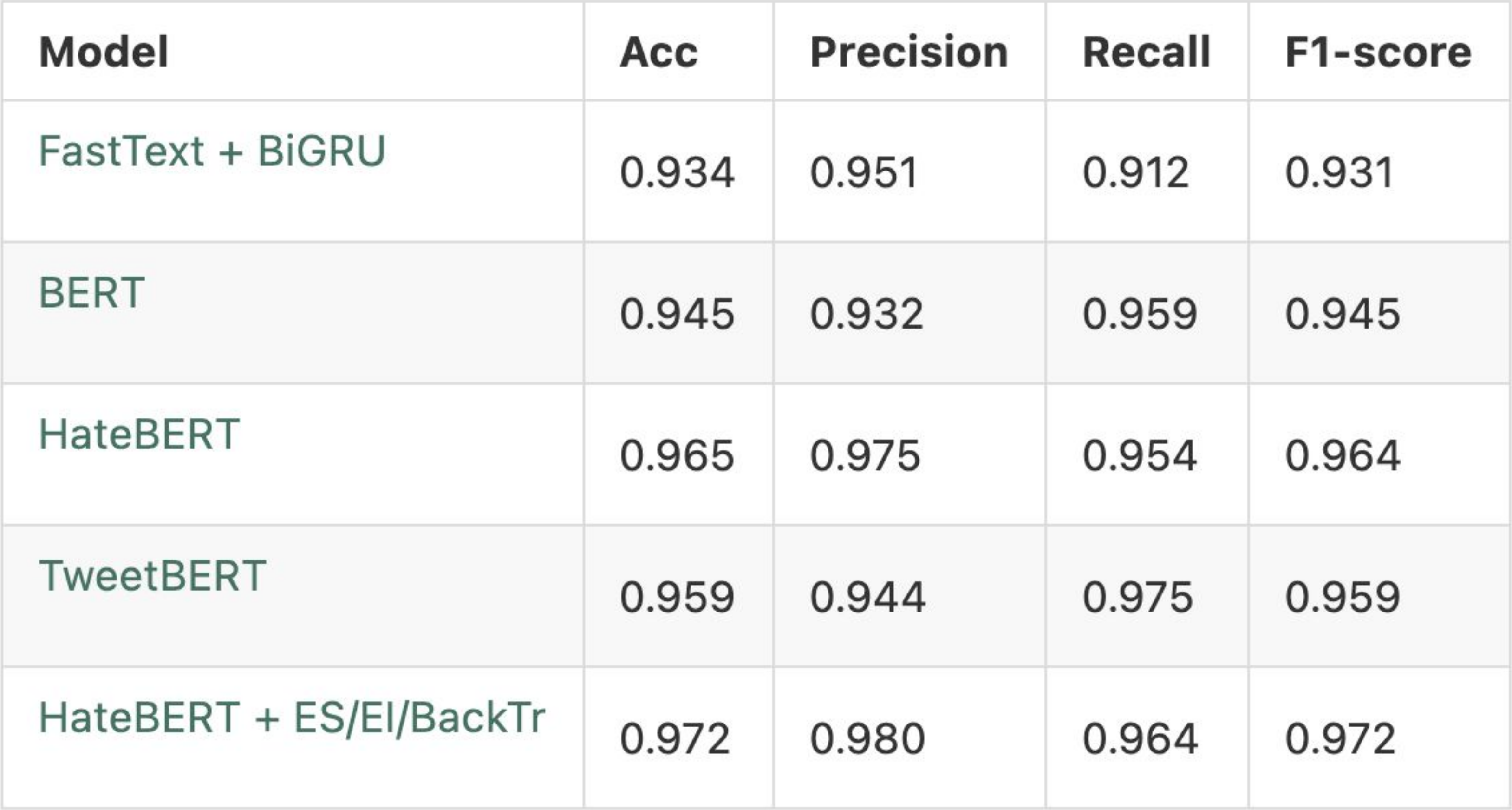
BERT / HateBERT / Roberta



BERT


NSP
Mask LM
Mask LM
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
...
T_M'
BERT
$E_{[CLS]}$
E_1
...
E_N
$E_{[SEP]}$
E_1'
...
E_M'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
...
T_M'
BERT
$E_{[CLS]}$
E_1
...
E_N
$E_{[SEP]}$
E_1'
...
E_M'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

Results



Model	Acc	Precision	Recall	F1-score
FastText + BiGRU	0.934	0.951	0.912	0.931
BERT	0.945	0.932	0.959	0.945
HateBERT	0.965	0.975	0.954	0.964
TweetBERT	0.959	0.944	0.975	0.959
HateBERT + ES/EI/BackTr	0.972	0.980	0.964	0.972

Basic autocomplete

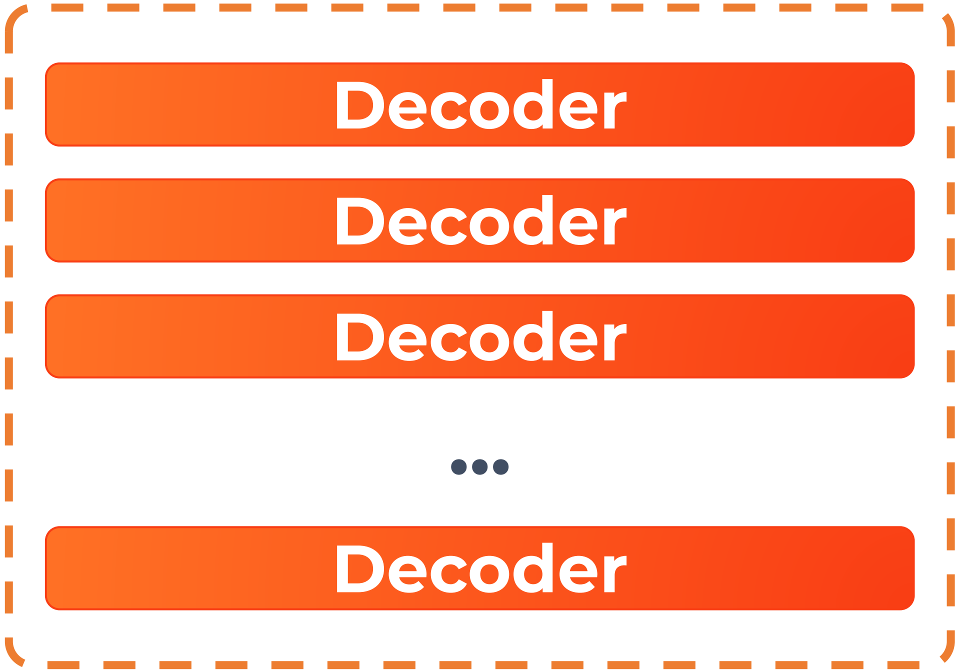
$$P(m_0, m_1, \cdots m_N | c_0, c_1, \cdots c_T) = \prod_{i=1}^{N} P(m_i | c_0, c_1, \cdots c_{i-1})$$

Predict the most likely sequence of tokens given a preceding code context

Transformers for code generation

Encoder	Decoder	Encoder + Decoder
Classification	Auto-complete	Translate English-Code
BERT	GTP	BART
		T5

GPT (Generative Pre-Training)



Auto-complete

I am ... → Jojo

Translation

I am <to_fr> je ... → suis

Summarization

Bla bla bla <summarize> ... → Bla

GTP

IntelliCode

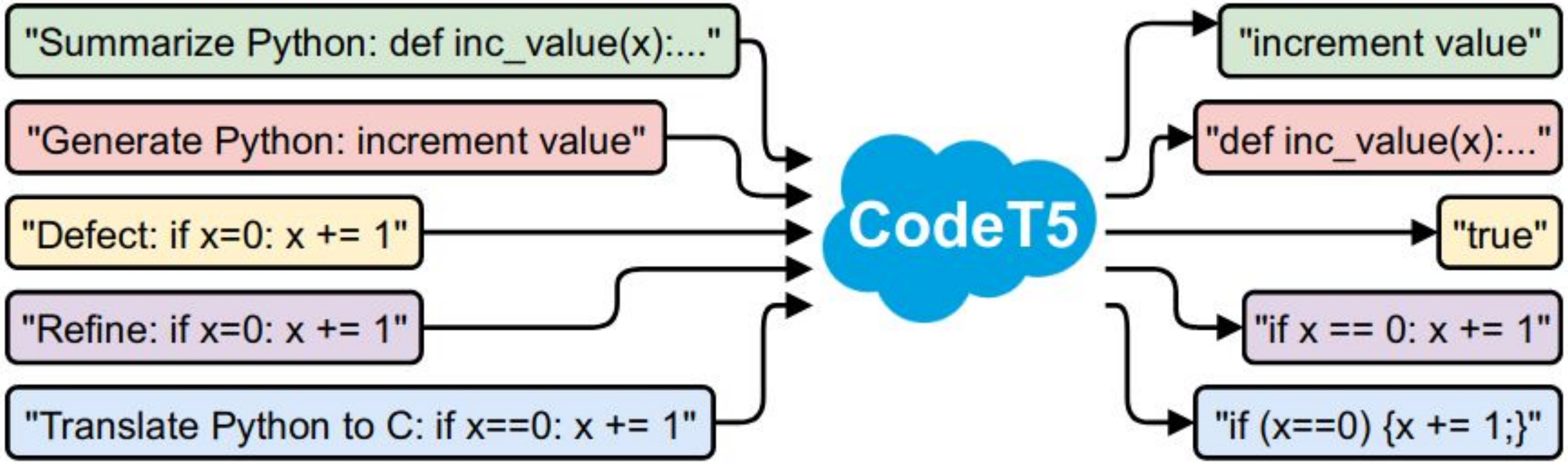
- Based on GTP-2
- 9 Languages
- Dataset : GitHub
- <CHAR_LIT>, <COMMENT>, ...
- Prefix Tree Caching

OpenAI Codex (GitHub Copilot)

- Based on GTP-3
- 12 Languages
- 12B parameters (GPT-3 : 175B)
- Dataset : GitHub

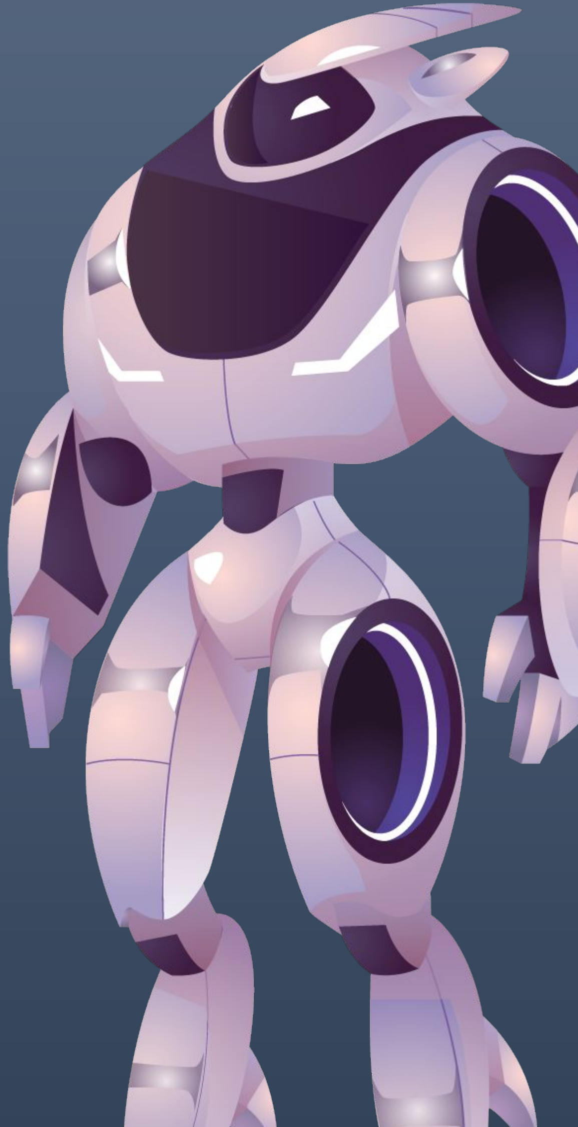
code-T5

Text-To-Text Transfer Transformer



https://bit.ly/lse-winter-transformers

Conclusion



Thanks !