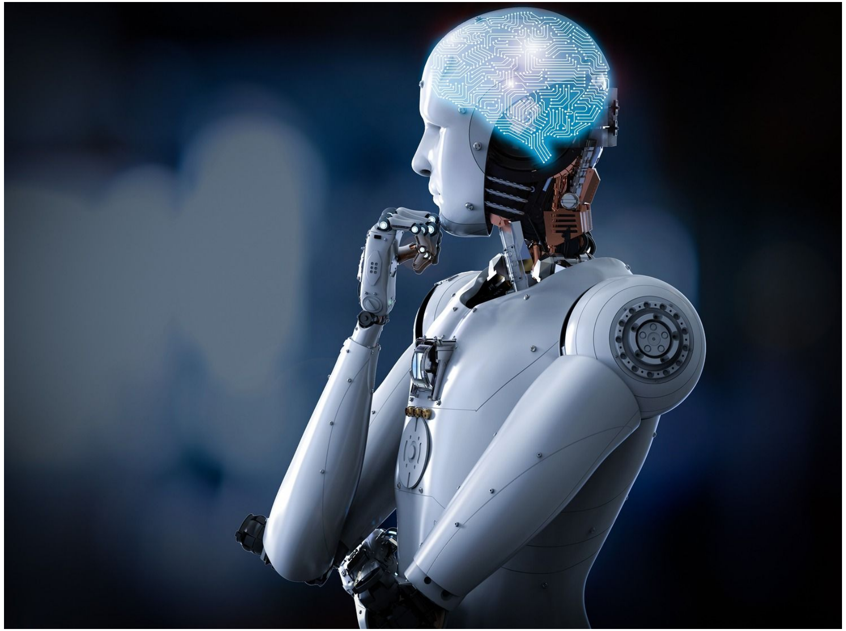
كلية العلوم بقابس
Faculté des Sciences de Gabès

Hatem Bettahar Research Lab
ResCoMath

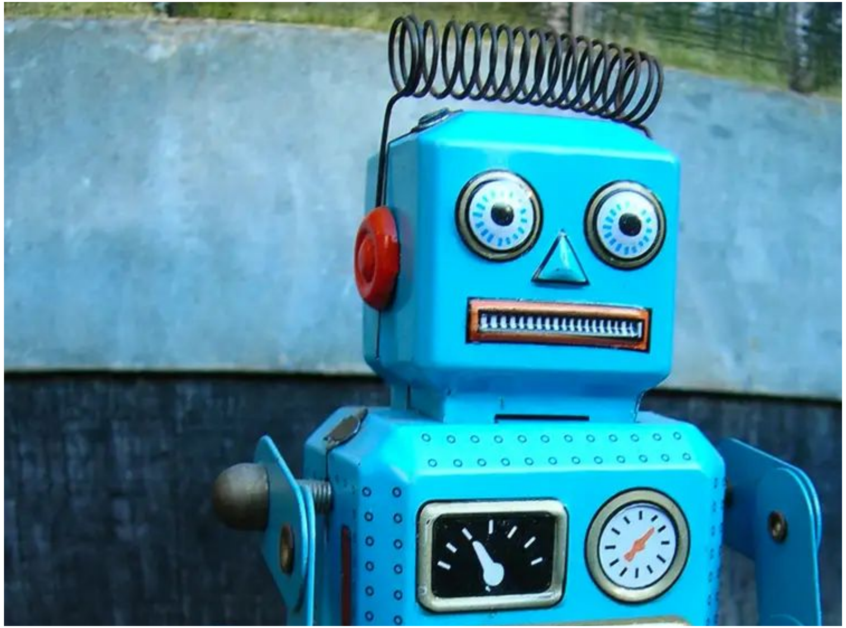
Introduction to Large Language Models

Eya Ben Charrada

Too smart or too dumb?



ChatGPT has passed medical, law and business schools exams



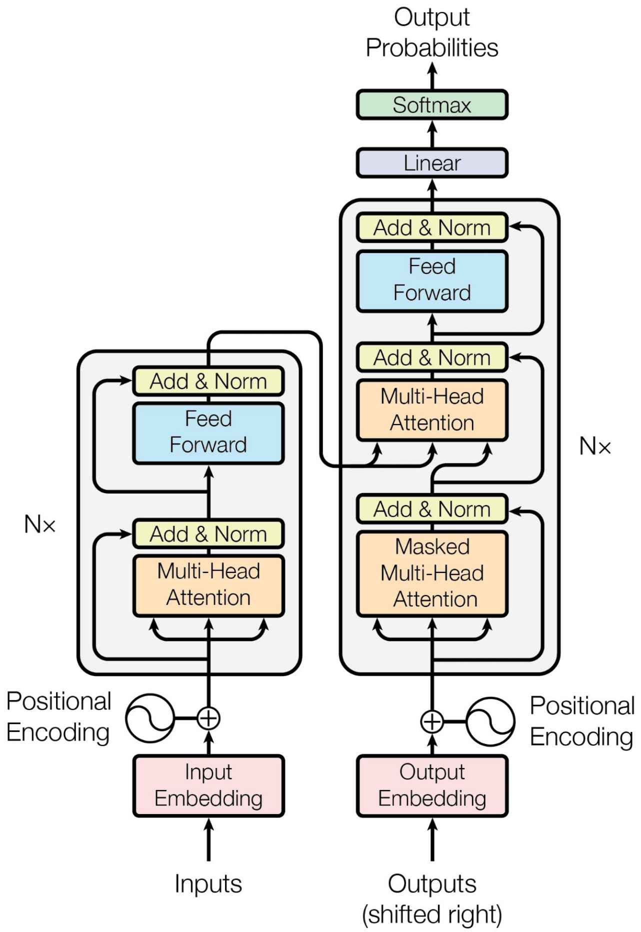
Fails to answer questions that a 5-years-old child can answer

How does it work?

Famous google paper (2017)

Attention Is All You Need

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



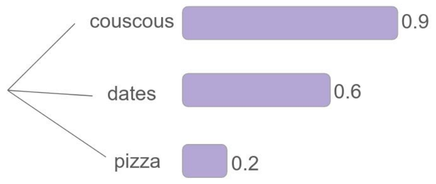
Goal

Understand how transformer models work :

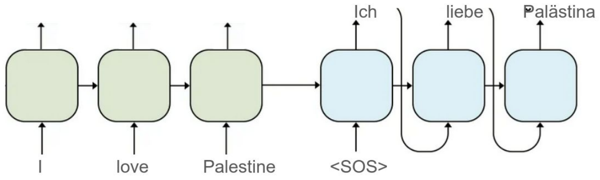
Explore the hidden meaning behind mathematical equations

Outline

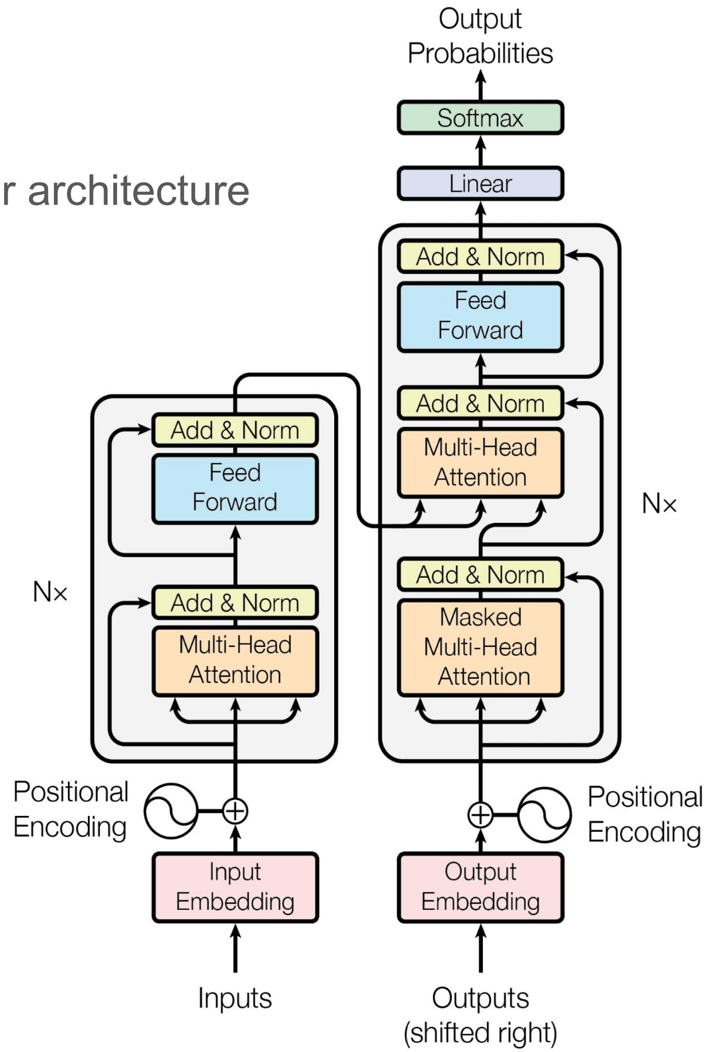
What is a language model



Before the transformer

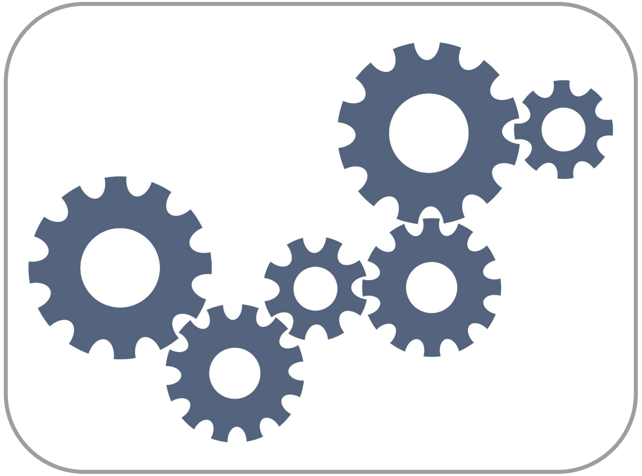


Transformer architecture



Language Model

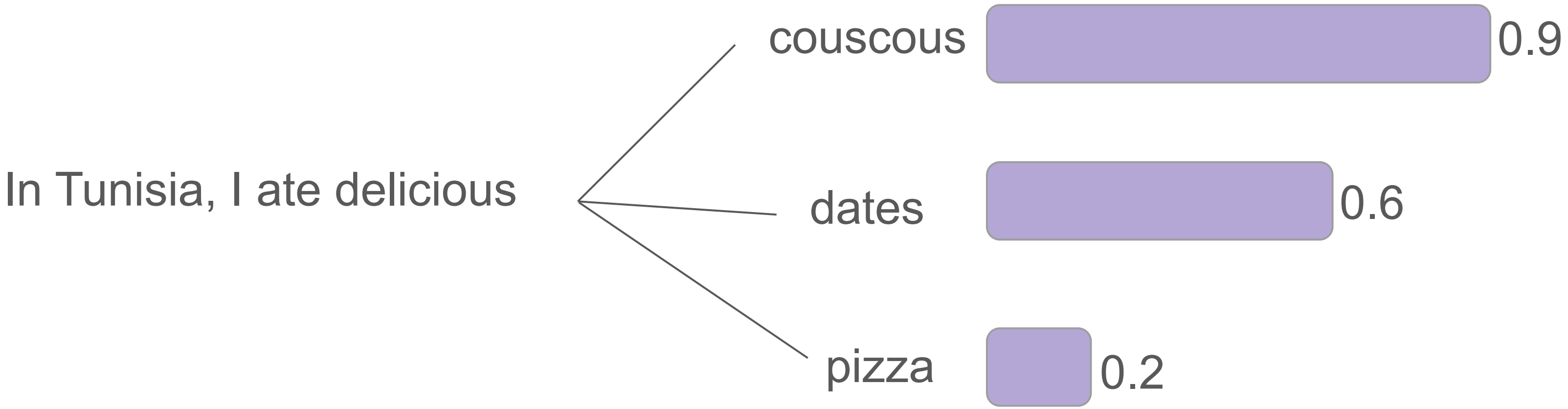
In Tunisia, I ate delicious →



→ ?

Translation, summarisation, question answering…

Language Model



Large Language Model

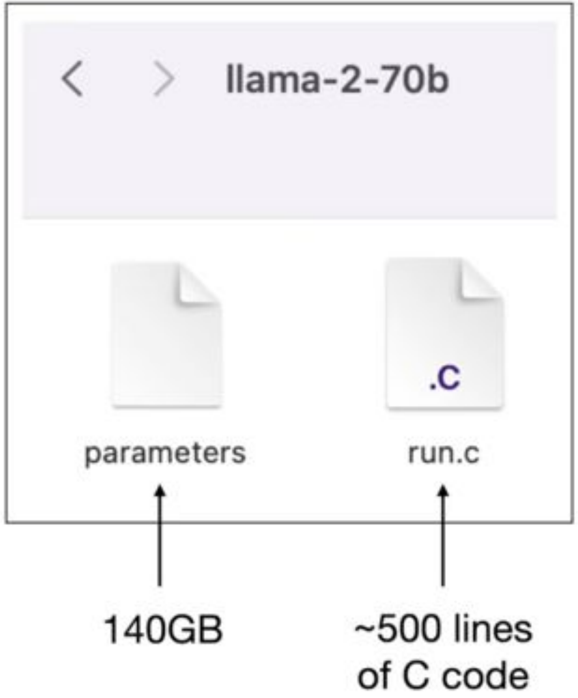


Image credit: Andrej Karpathy

Training costs

10 TB of Text (scraped from the internet)

6000 GPUs for 12 days ~2M$*

* Data For Llama 2 - 70 B according to Karpathy

Parameters

Calculated based on training

Parameters are not interpretable or understandable by human

No one knows how they collaborate

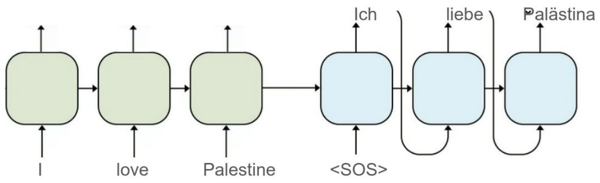
How do language models work?

In Tunisia, I ate delicious →  → couscous

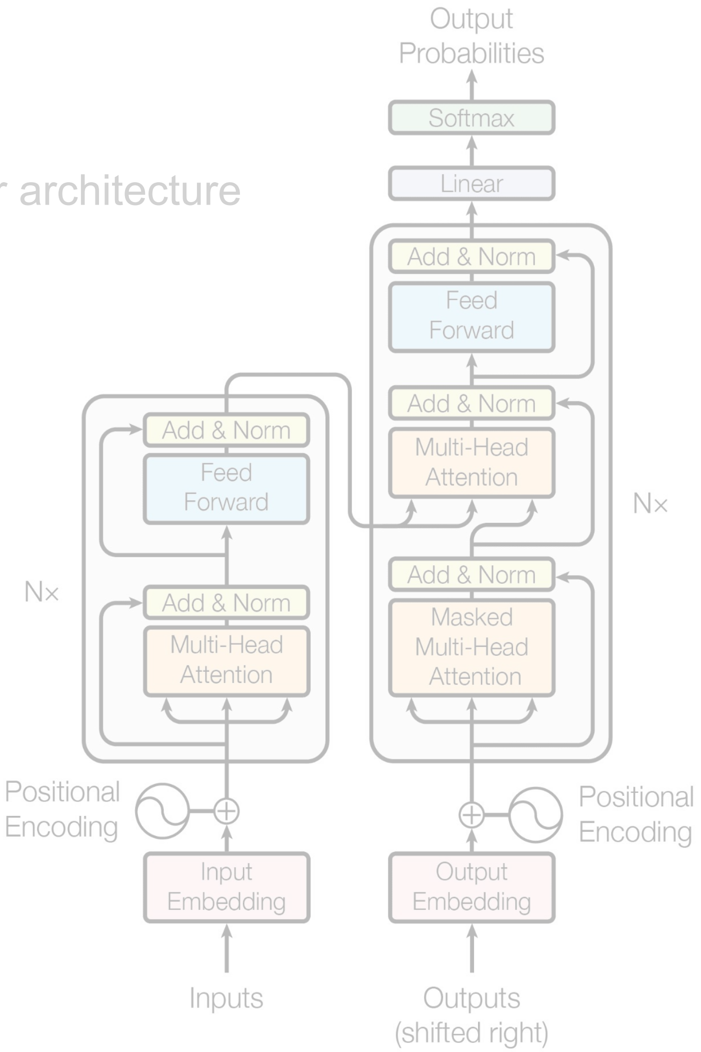
Outline

What is a language model

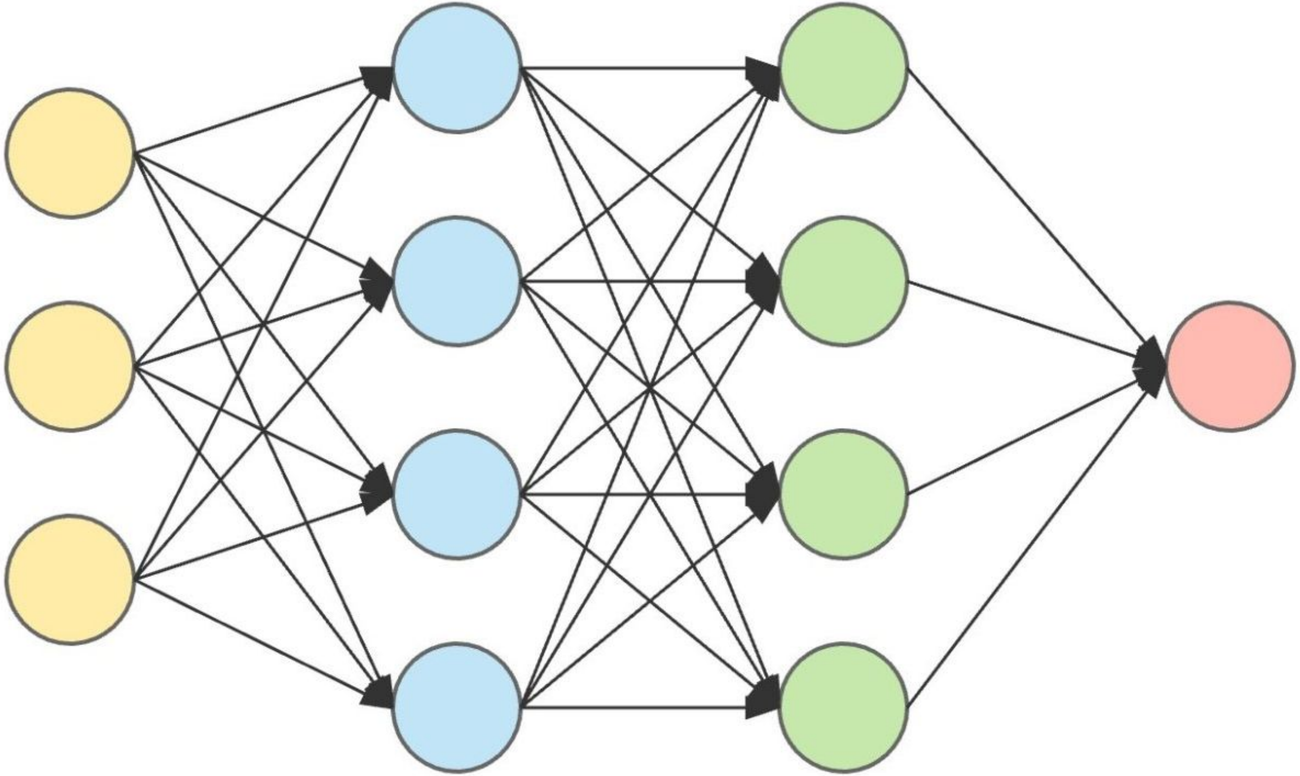
Before the transformer



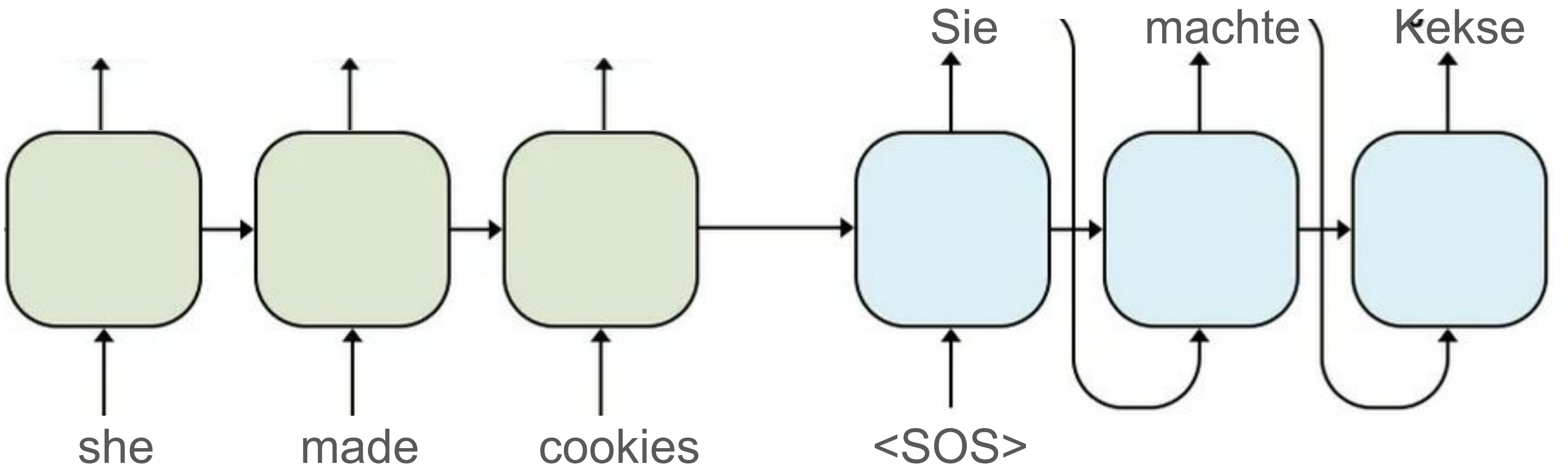
Transformer architecture



Neural network

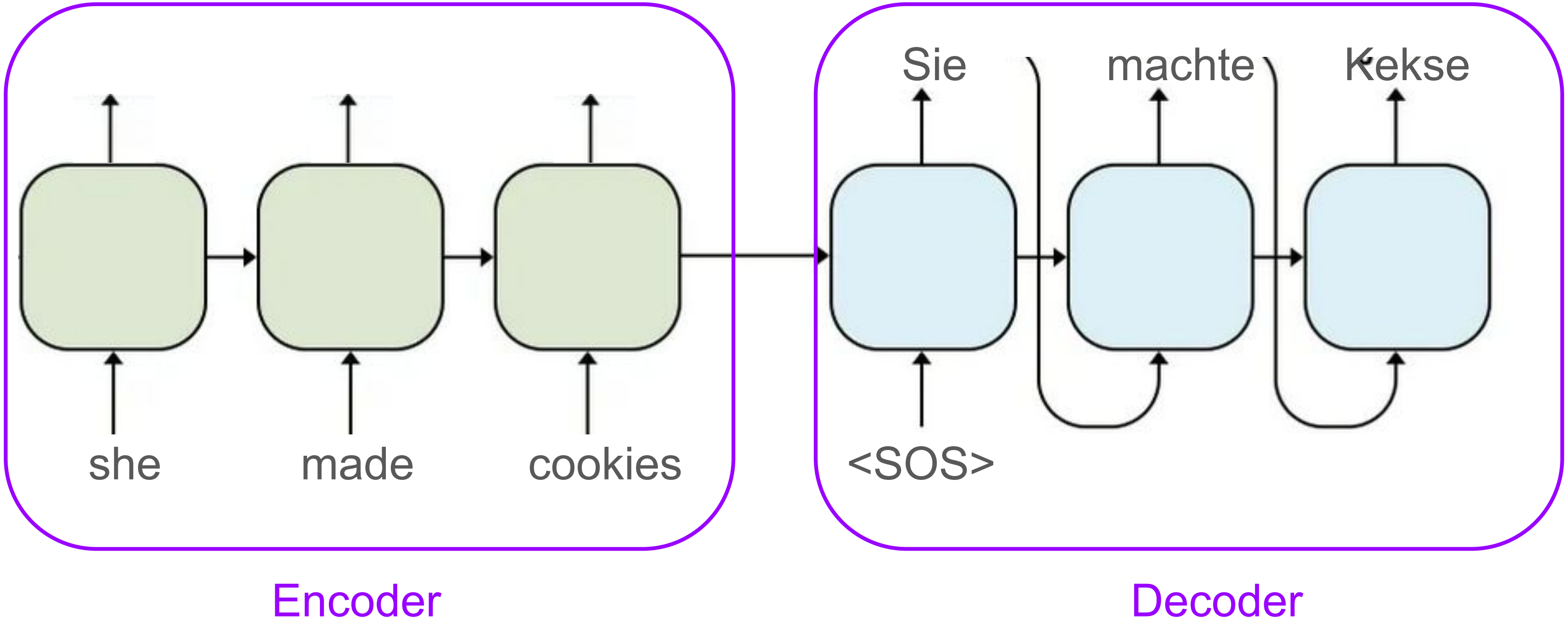


Sequence to sequence models - Translation

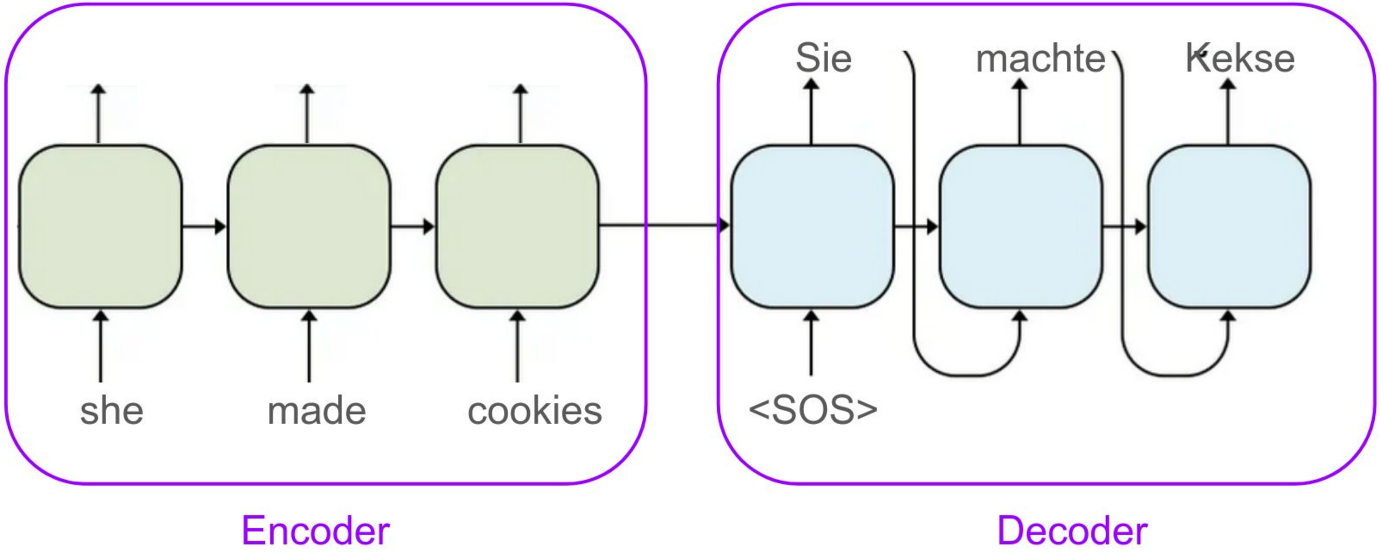


Takes a sequence as input and generated another sequence
Example: Recurrent neural network (RNN), Gated Recurrent Units (GRU), Long-short-term memory (LSTM)

Sequence to sequence models



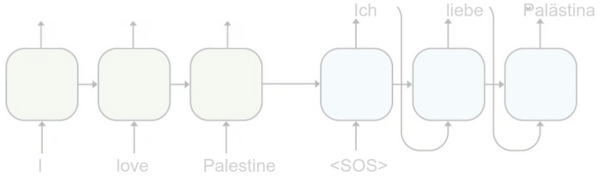
Sequence to sequence models



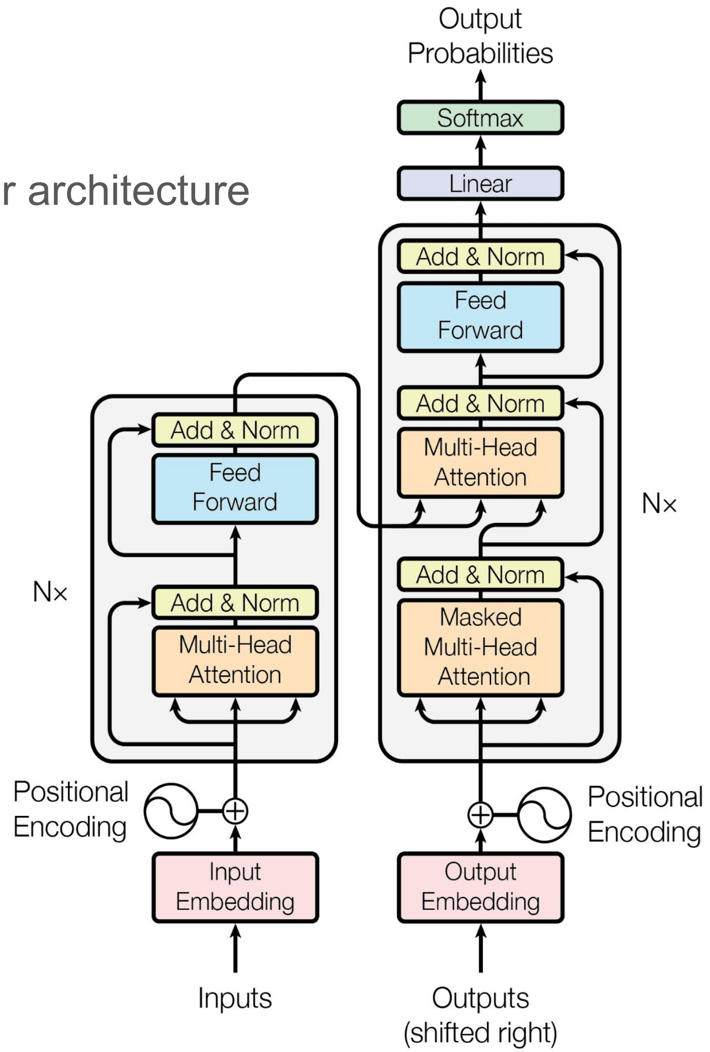
These models process the sequence word by word. This makes them very expensive to train.

What is a language model

Before the transformer

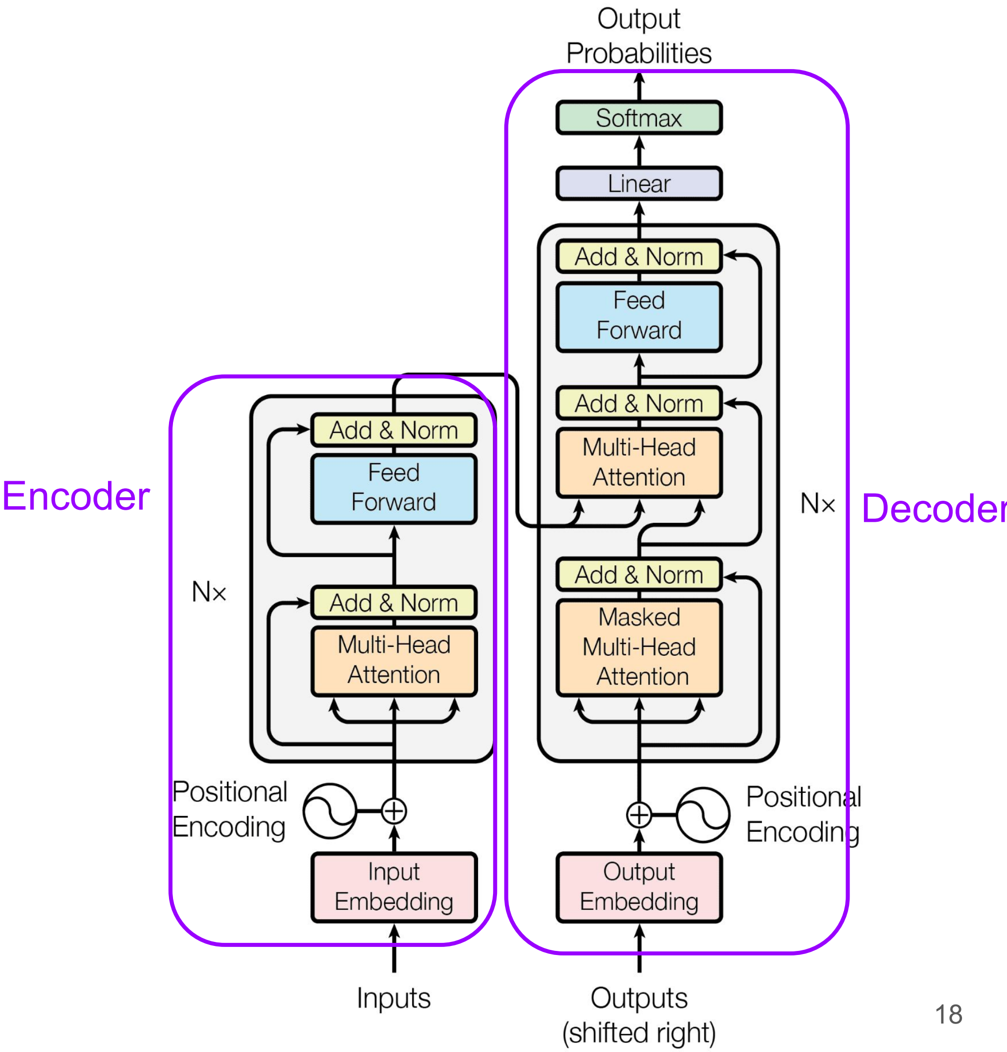


Transformer architecture

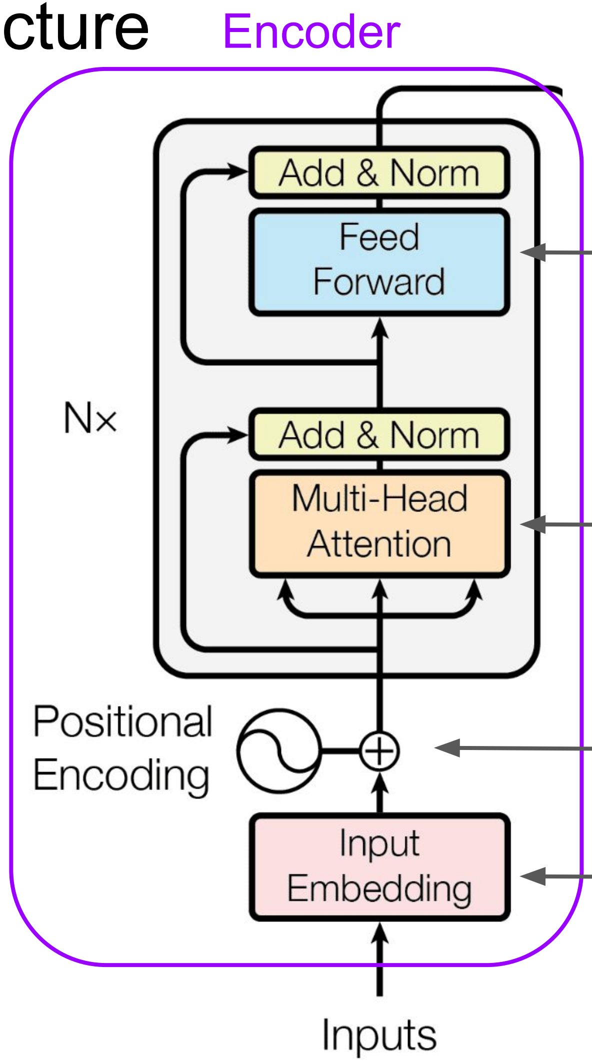


Transformer architecture

"Attention is all you need"
(2017)



Transformer architecture



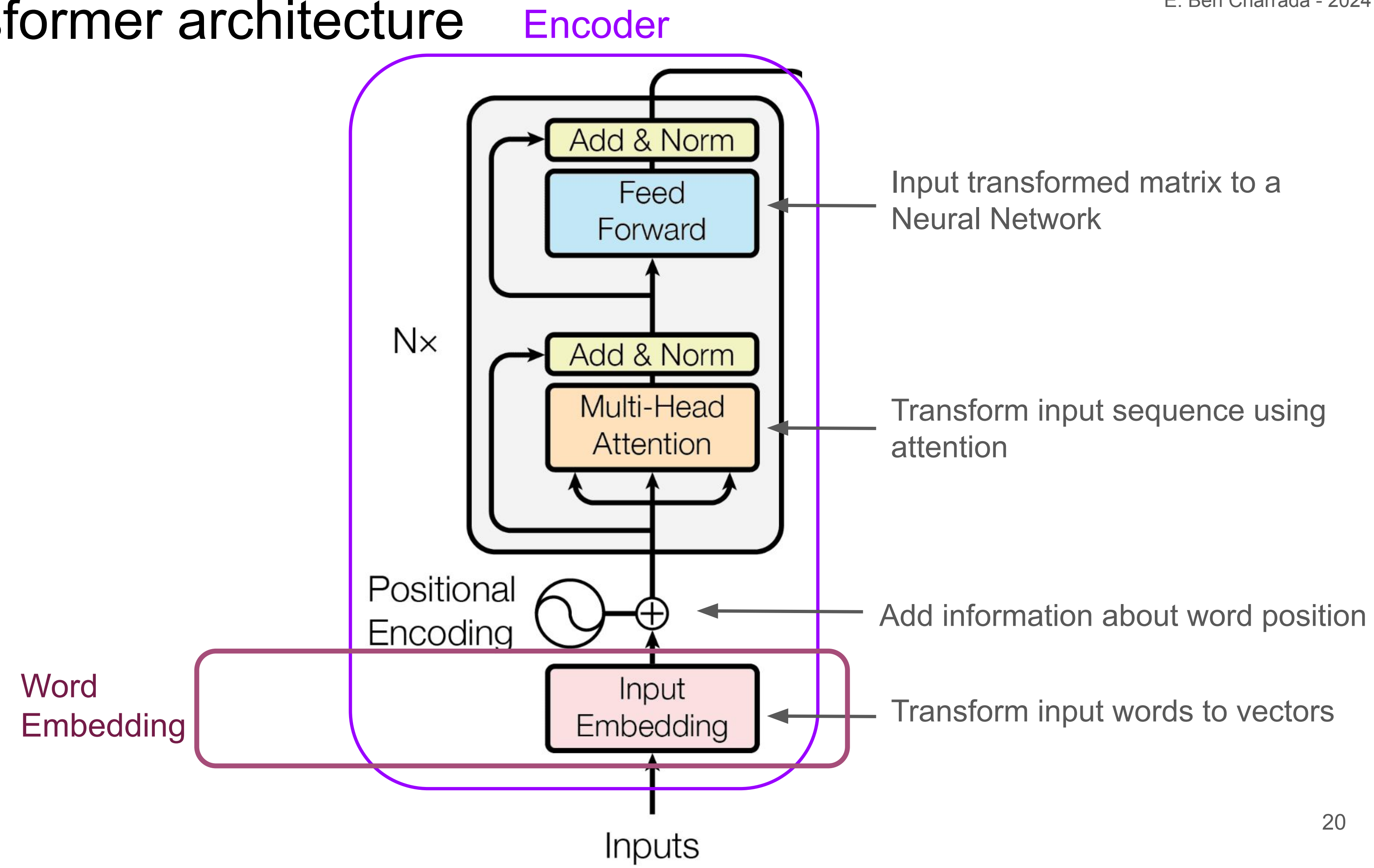
Input transformed matrix to a Neural Network

Transform input sequence using attention

Add information about word position

Transform input words to vectors

Transformer architecture



Sequence as Matrix

apple

3.4
5.2
-0.3
0.4

She made apple cake

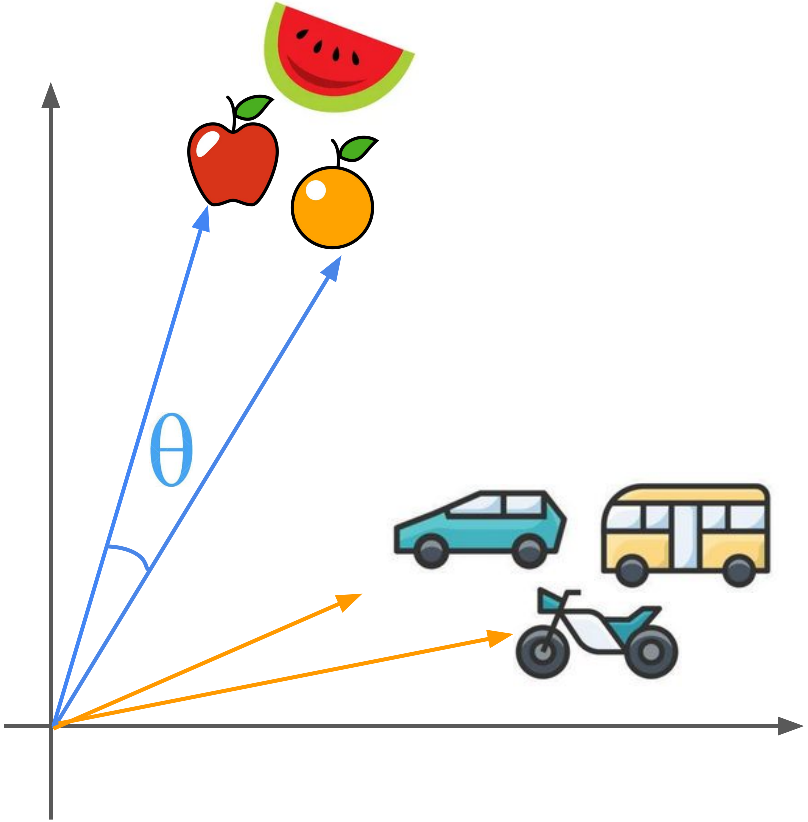
0.1	2.2	3.4	6.4
5.2	1.0	5.2	2.2
0.7	2.3	-0.3	5.3
2.4	-1.0	0.4	3.1

Word embedding

Words are represented as a vectors in a space of n dimensions

Similar words are closer to each other

Distance is usually calculated using cosine similarity: $\cos(\theta)$



Word embedding

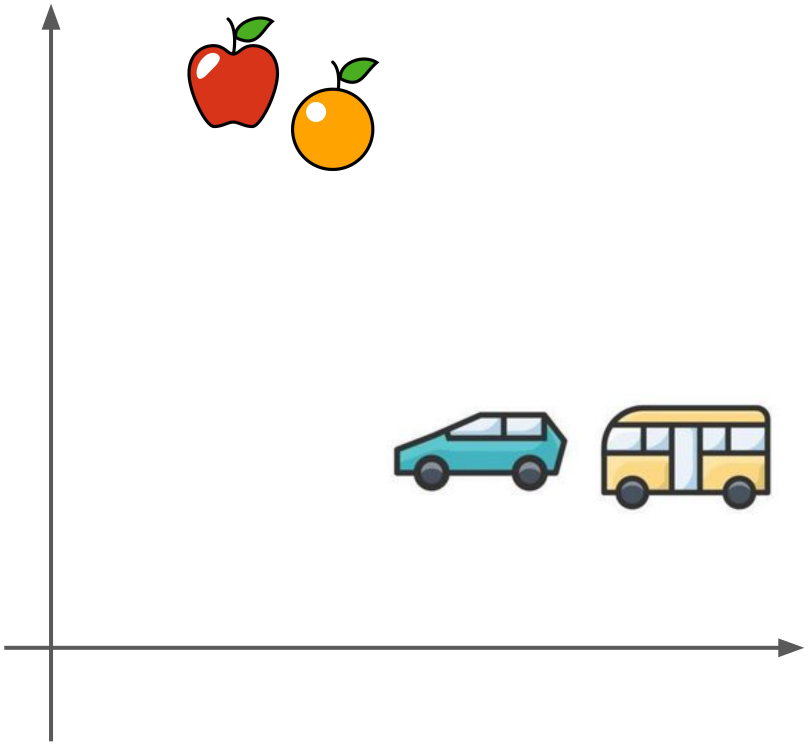
I made an apple juice

I made orange juice

I made a fruit salad with apples and oranges

I go to school by car

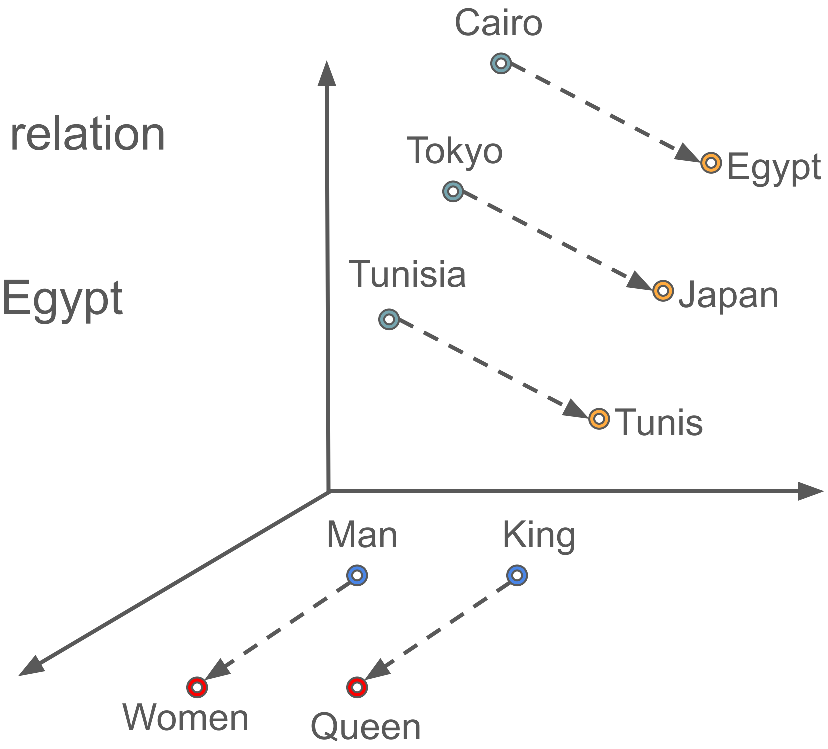
The school bus arrived on time



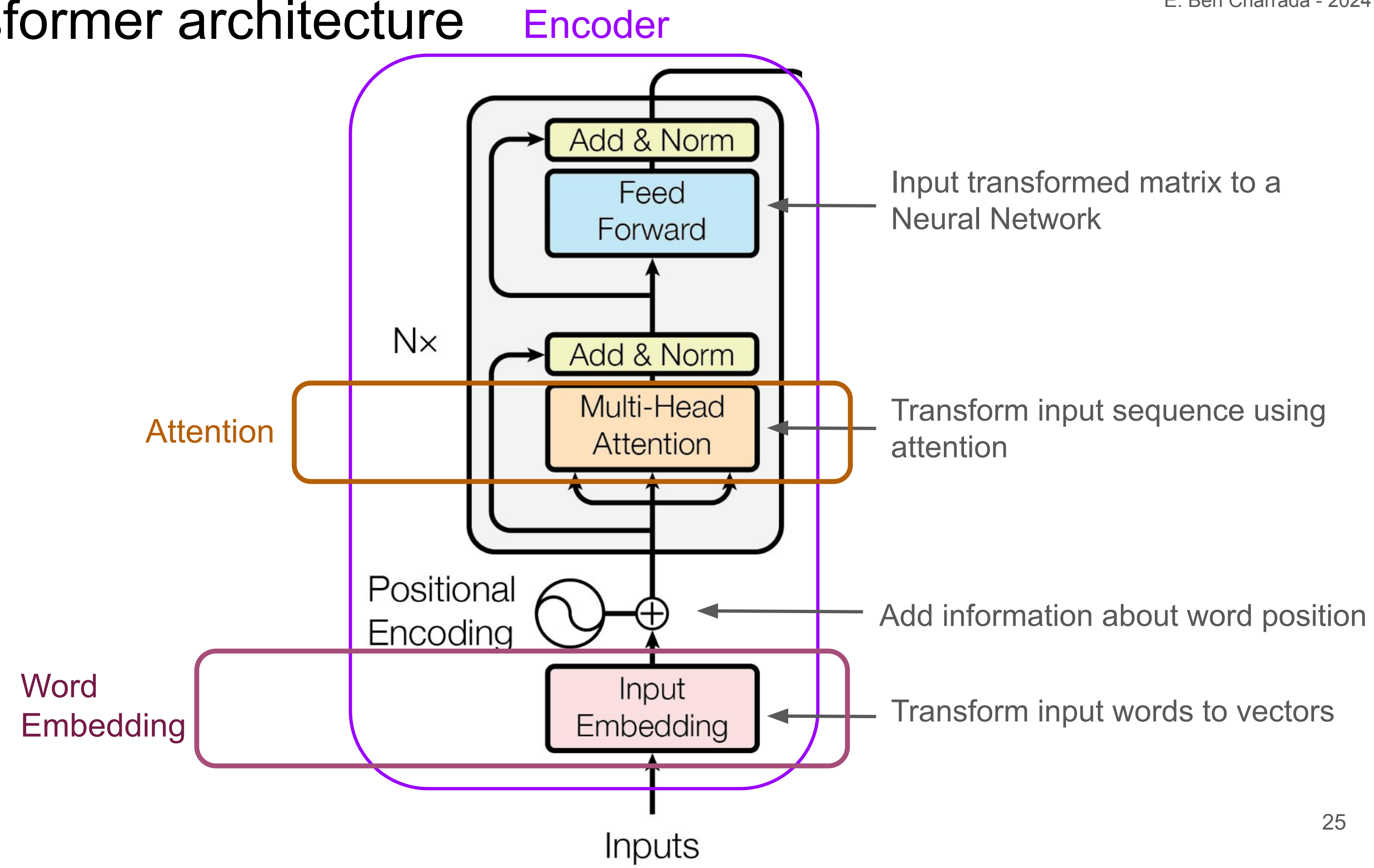
Word embedding

The distance between words reflects the relation between the words

Example: Tokyo to Japan is like Cairo to Egypt

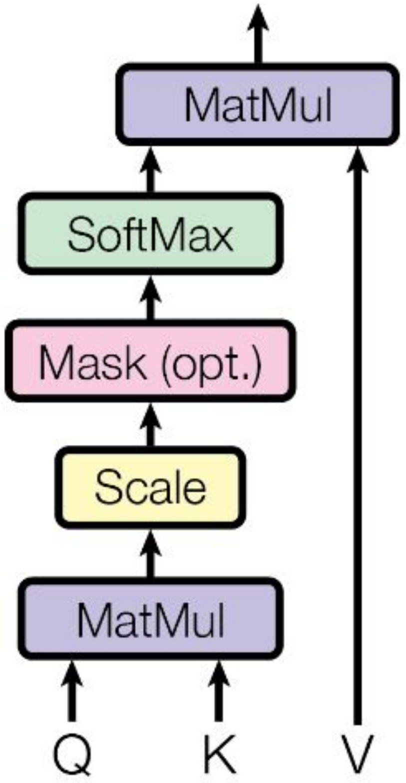


Transformer architecture



Attention mechanism

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



The math is too simple.

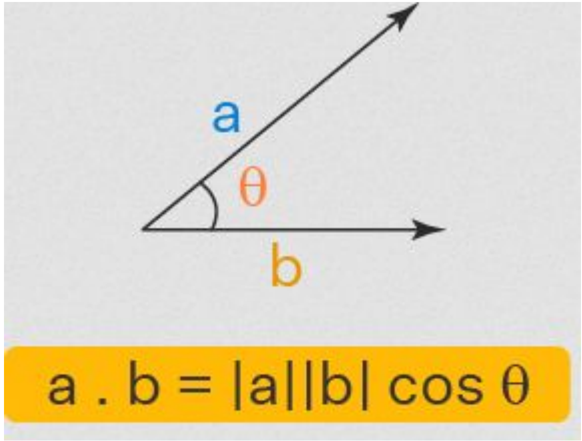
But wait…

What's the meaning of multiplying the input 3 times?

Ideas?

Attention mechanism - Explained

The value of the dot product depends on the cosine distance between the vectors



So similar words will have a higher dot product

Sequence: she made apple cake

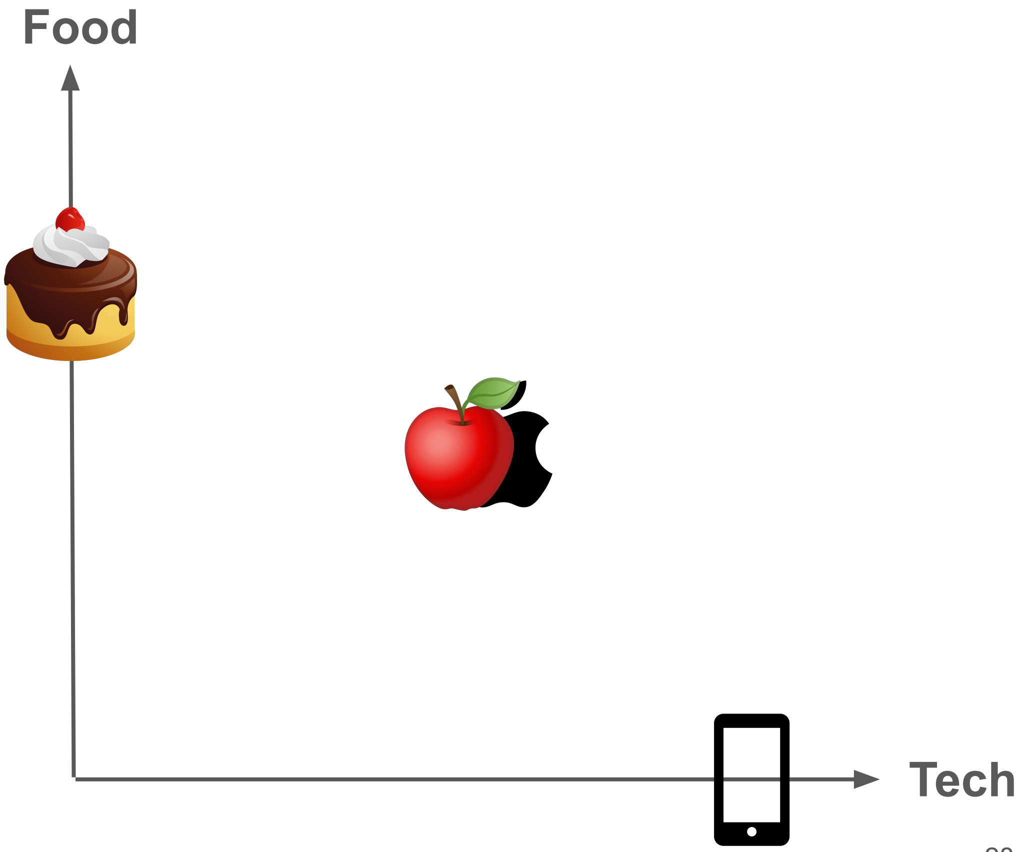
Q

	Food	Tech	Other
cake	3	0	0
apple	2	2	0
made	0	0	2
she	0	0	3

●

Q^T

cake	apple	made	she
3	2	0	0
0	2	0	0
0	0	2	3



Example inspired by Serrano.Academy

First dot product

Cookies and *dates* have a higher dot product than *dates* and *she*

The multiplication gives higher values to similar words in the input sequence

	cookies	apple	made	she
cookies	9	6	0	0
dates	6	8	0	0
made	0	0	4	6
she	0	0	6	9

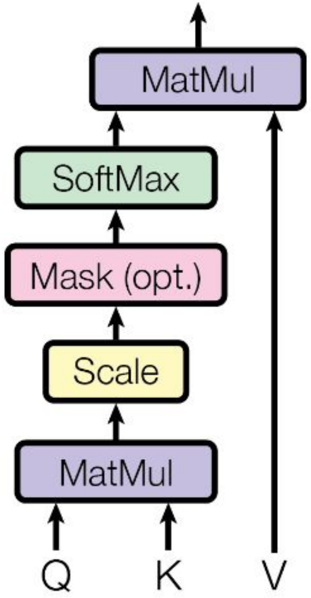
Scaling & Softmax

Scaling:

For large values of d_k, the dot products grow large in magnitude. To counteract this effect, the dot product is scaled by $1/\sqrt{d_k}$

Softmax: $$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Returns values between 0 and 1 while preserving the order of the elements



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

First dot product

Orange and *apple* have a higher dot product than *apple* and *an*

The multiplication gives higher values to similar words in the input sequence

	cookies	apple	made	she
cookies	0.8	0.2	0	0
apple	0.2	0.8	0	0
made	0	0	0.6	0.3
she	0	0	0.4	0.5

Second dot product

	cake	apple	made	she
cake	0.8	0.2	0	0
apple	0.2	0.8	0	0
made	0	0	0.6	0.3
she	0	0	0.4	0.5

●

	Food	Tech	Other
cake	3	0	0
apple	2	2	0
made	0	0	2
she	0	0	3

Second dot product

	cake	apple	made	she
cake	0.8	0.2	0	0
apple	0.2	0.8	0	0
made	0	0	0.6	0.3
she	0	0	0.4	0.5

●

	Food	Tech	Other
cake	3	0	0
apple	2	2	0
made	0	0	2
she	0	0	3

=

cake	2.8	0.4	
apple			
made			
she			

Second dot product

	cake	apple	made	she
cake	0.8	0.2	0	0
apple	0.2	0.8	0	0
made	0	0	0.6	0.3
she	0	0	0.4	0.5

●

	Food	Tech	Other
cake	3	0	0
apple	2	2	0
made	0	0	2
she	0	0	3

=

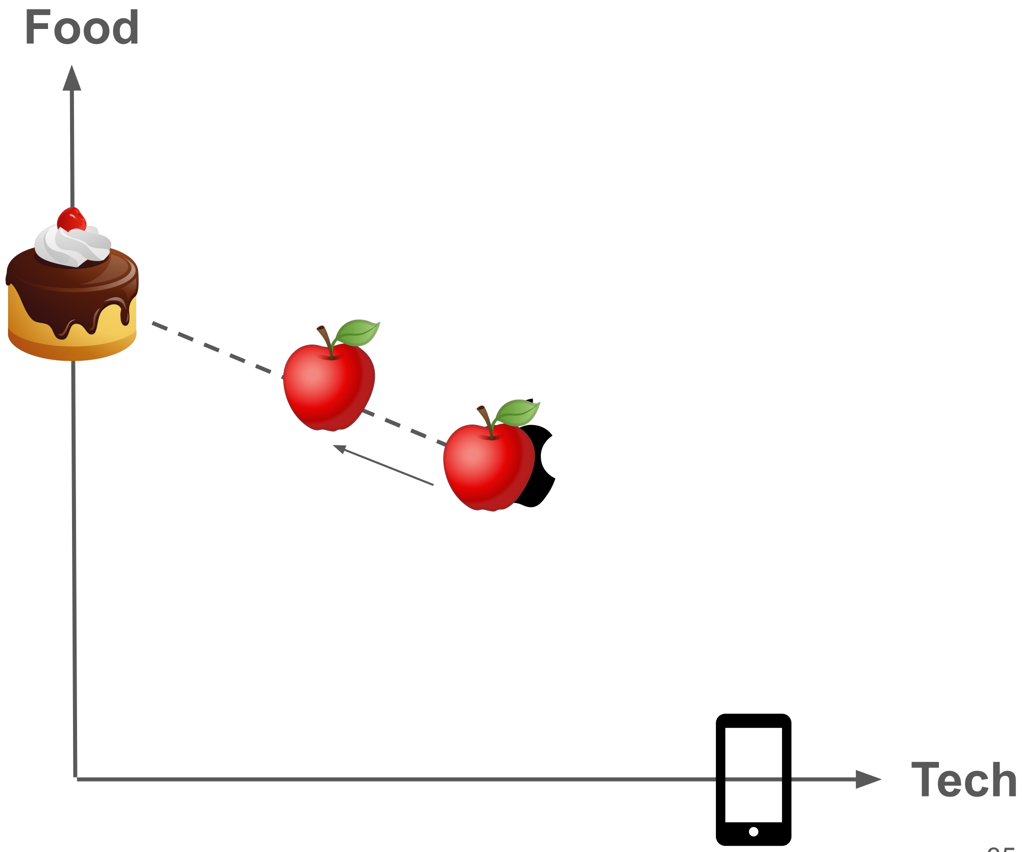
cake	2.8	0.4	0
apple	2.2	1.6	0
made	0	0	2.1
she	0	0	2.3

Words are attracted to each other. Similar words will have more attraction effect on each other than non similar words.

Dot product for context awareness

She made apple cake

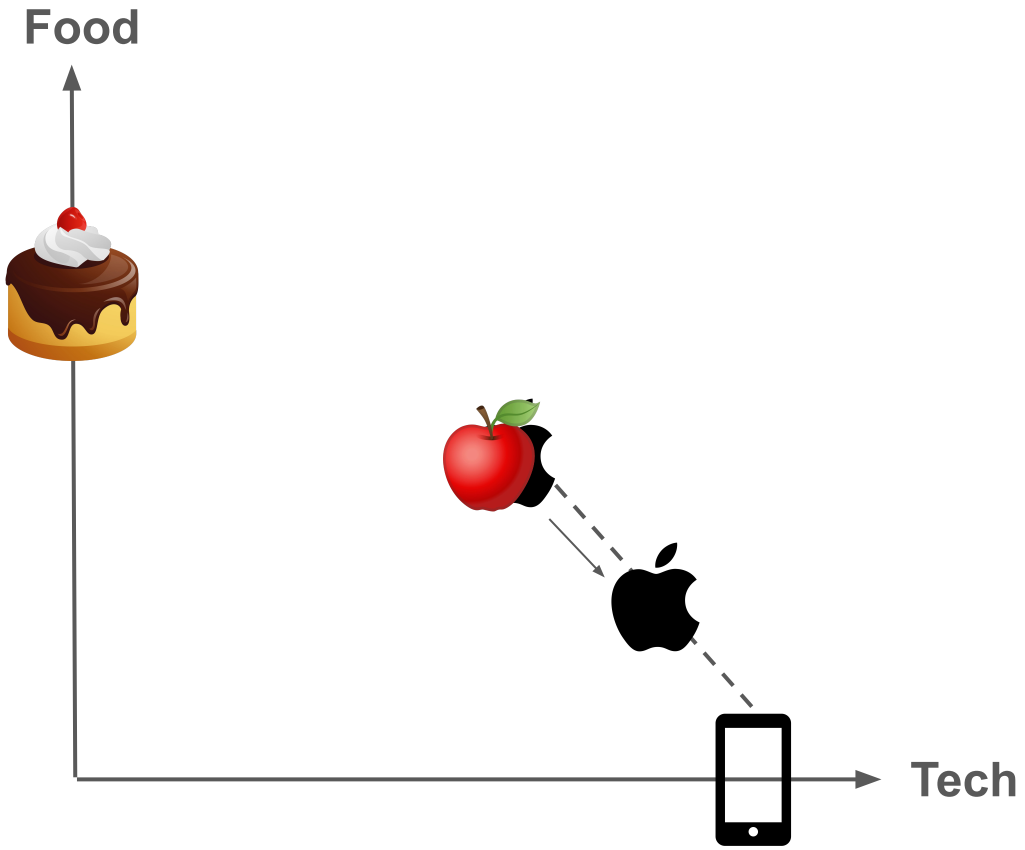
Using the matrix dot product will move "**apple**" toward "**cake**"



Example inspired by Serrano.Academy

Dot product for context awareness

new phone from apple



Example inspired by Serrano.Academy

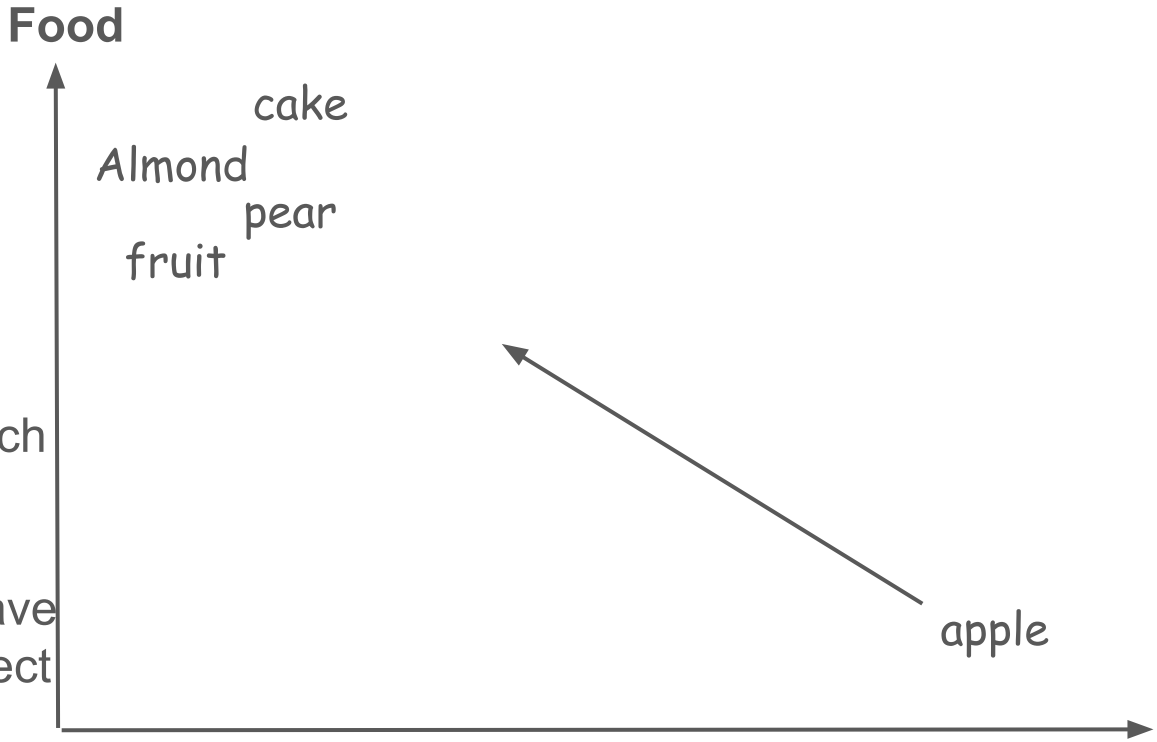
Dot product provides context awareness

She made a fruit cake with apples, pears and almond.

Gravity effect

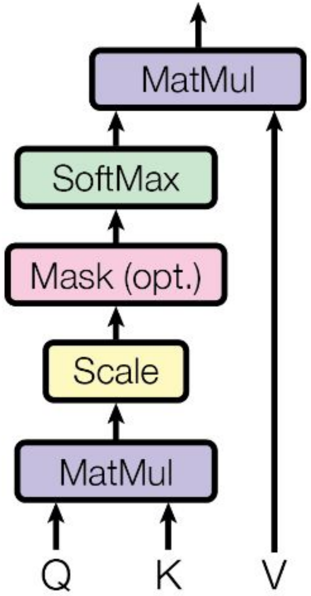
Similar words attract each other more

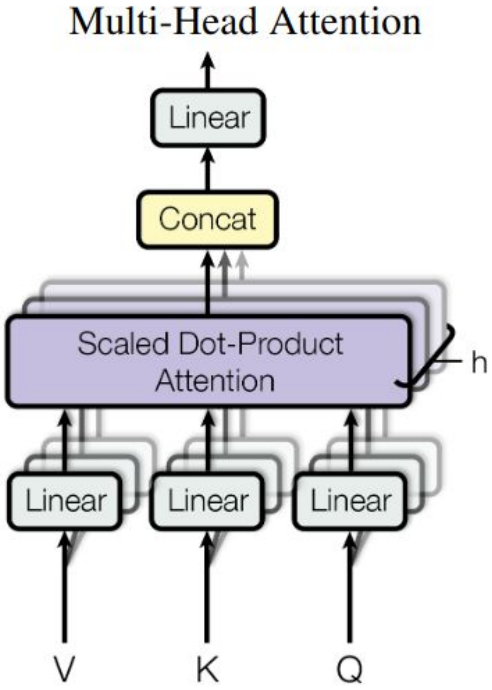
Cluster similar words have a stronger attraction effect

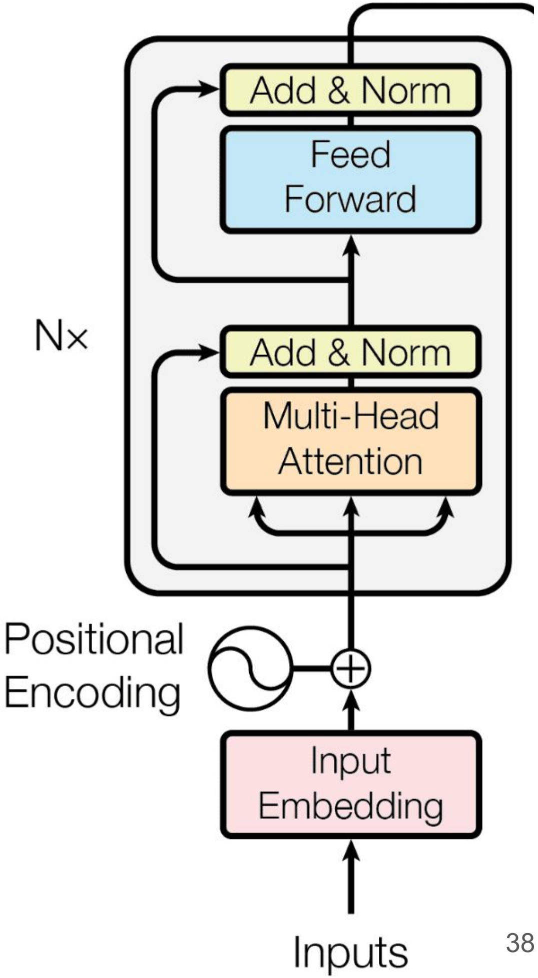


Example inspired by Serrano.Academy

Multi-Head Attention





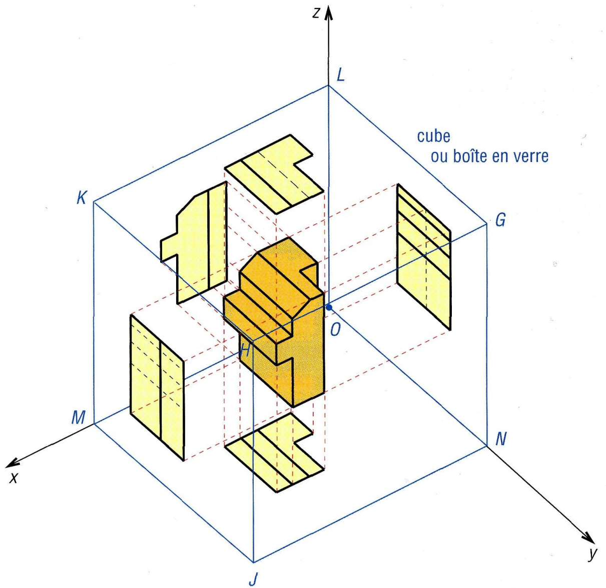


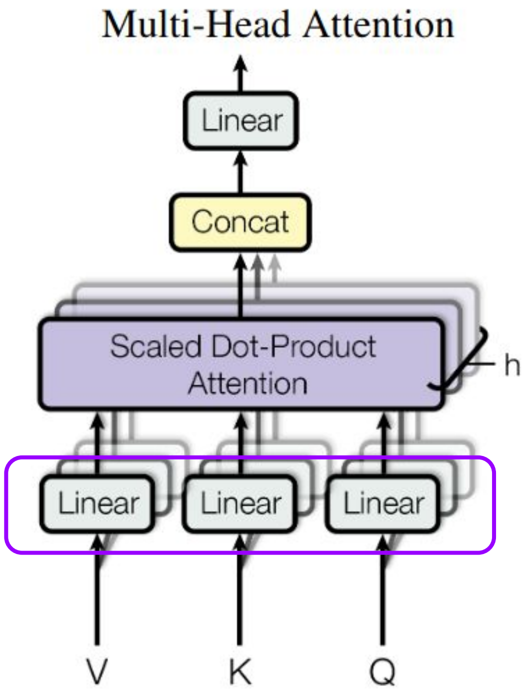
$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Projections

V, K and Q are the result of multiplying the sequence S with parameter matrices

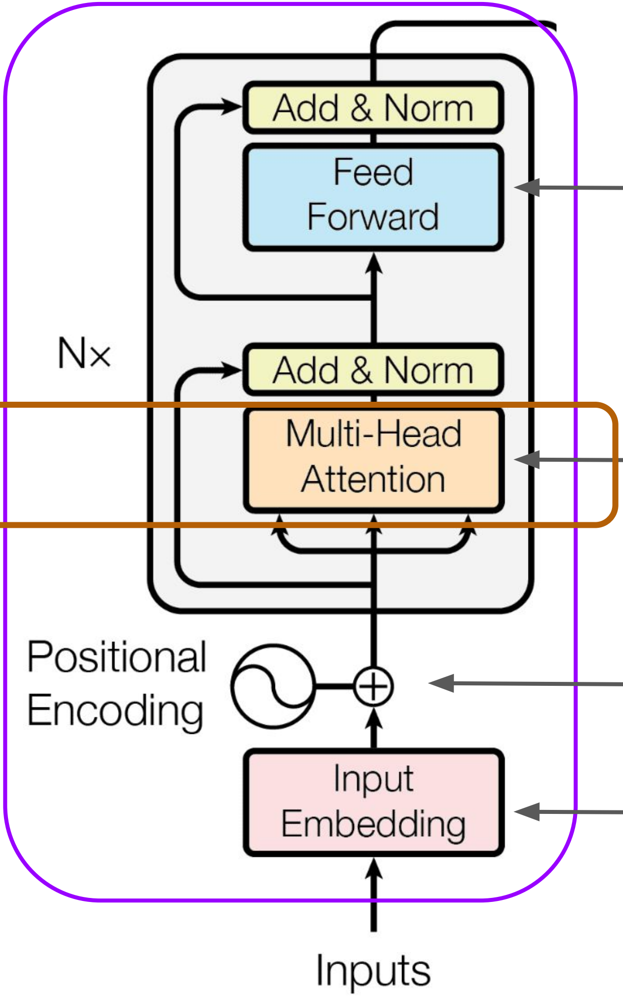
The multiplication is a linear projection of the sequence





Transformer architecture

Encoder



Attention

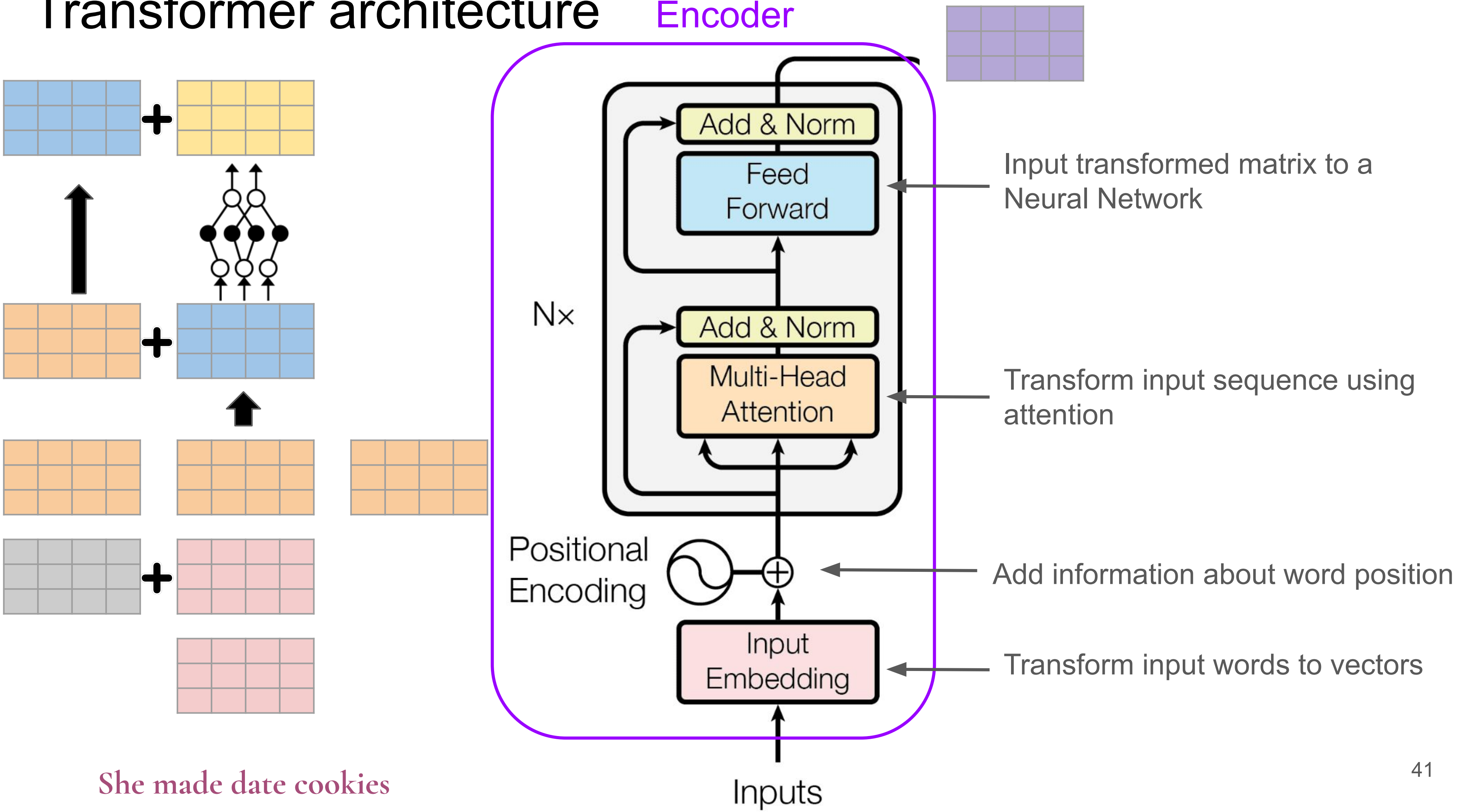
Input transformed matrix to a Neural Network

Transform input sequence using attention

Add information about word position

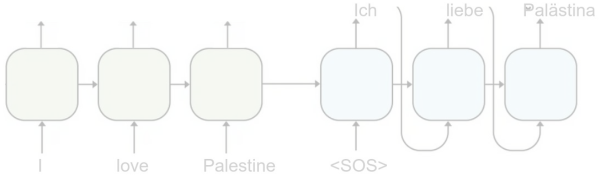
Transform input words to vectors

Transformer architecture

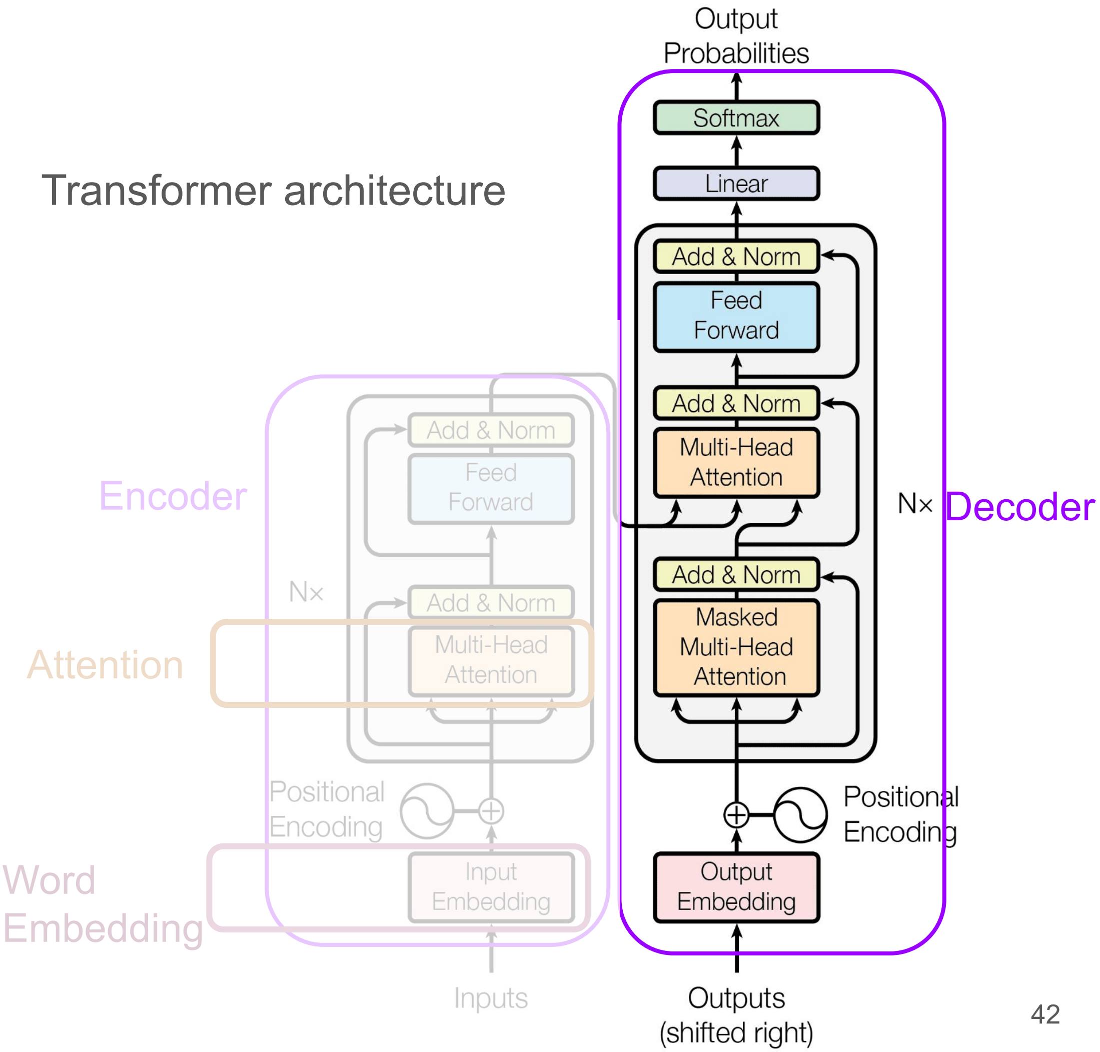


What is a language model

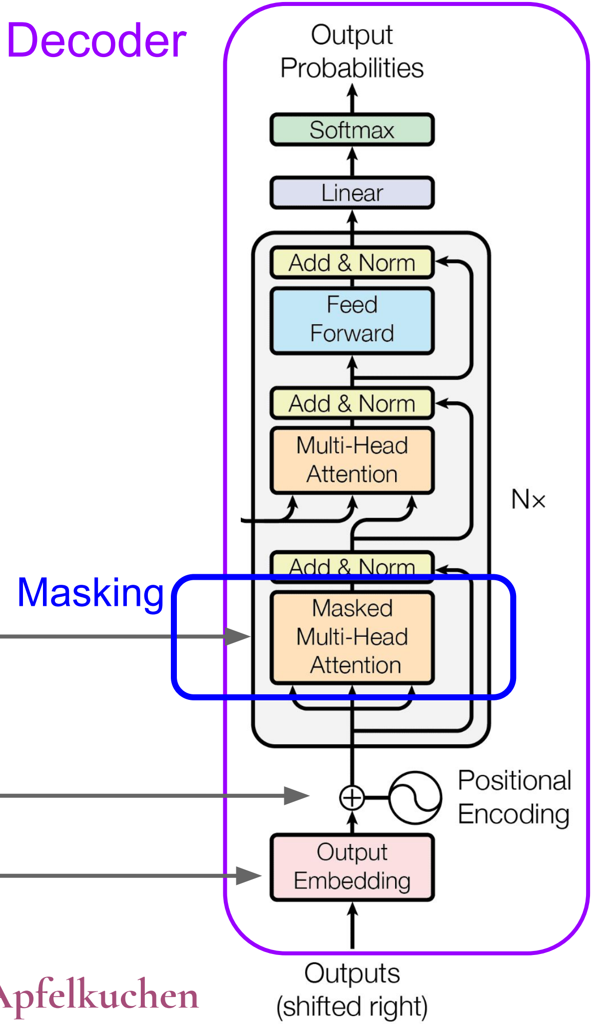
Before the transformer



Transformer architecture



Transformer architecture



Masked attention on output sequence

Add information about word position

Transform input words to vectors

<SOS> Sie machte Apfelkuchen

Masked attention

Mask out dependency with following words by setting it to -∞

Only previous words are taken into consideration

The softmax will transform the -∞ to 0

Allows processing the whole output sequence at once during learning

	<SOS>	Sie	machte	Apfelkuchen
<SOS>	0.8	-∞	-∞	-∞
Sie	0.1	0.7	-∞	-∞
machte	0	0.4	0.6	-∞
Apfelkuchen	0.2	0.1	0.4	0.5

Transformer architecture

Decoder

Projects the scores in the target vocabulary

Input transformed matrix to a Neural Network

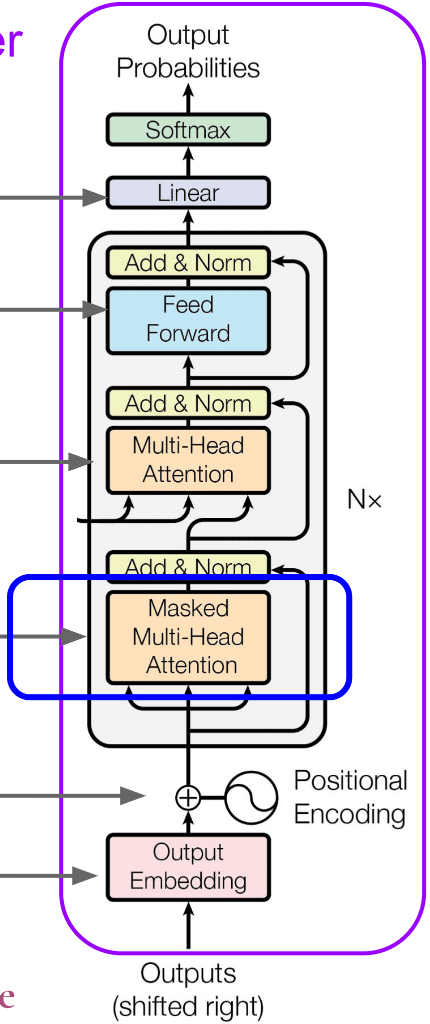
Attention with Q and K from Encoder

Masking

Masked attention on output sequence

Add information about word position

Transform input words to vectors



<SOS> Sie machte Dattelkekse

Costs of attention

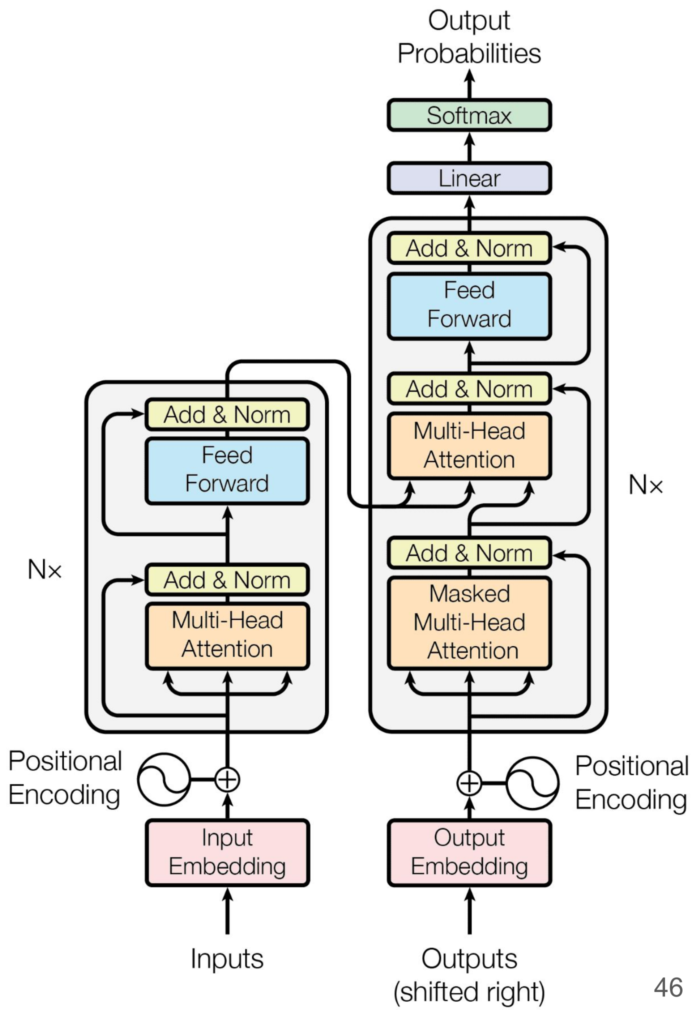
Learning:

Input/output sequences processed in one step

Inference:

Input sequence processed in one step

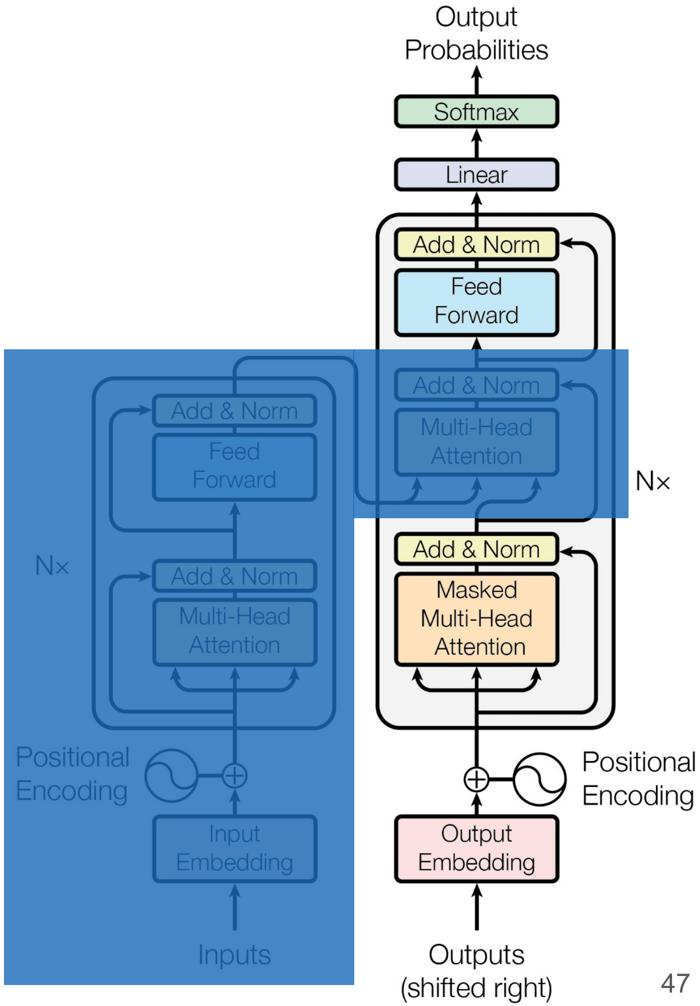
Output inferred token by token



GPT

Generative Pre-trained Transformer

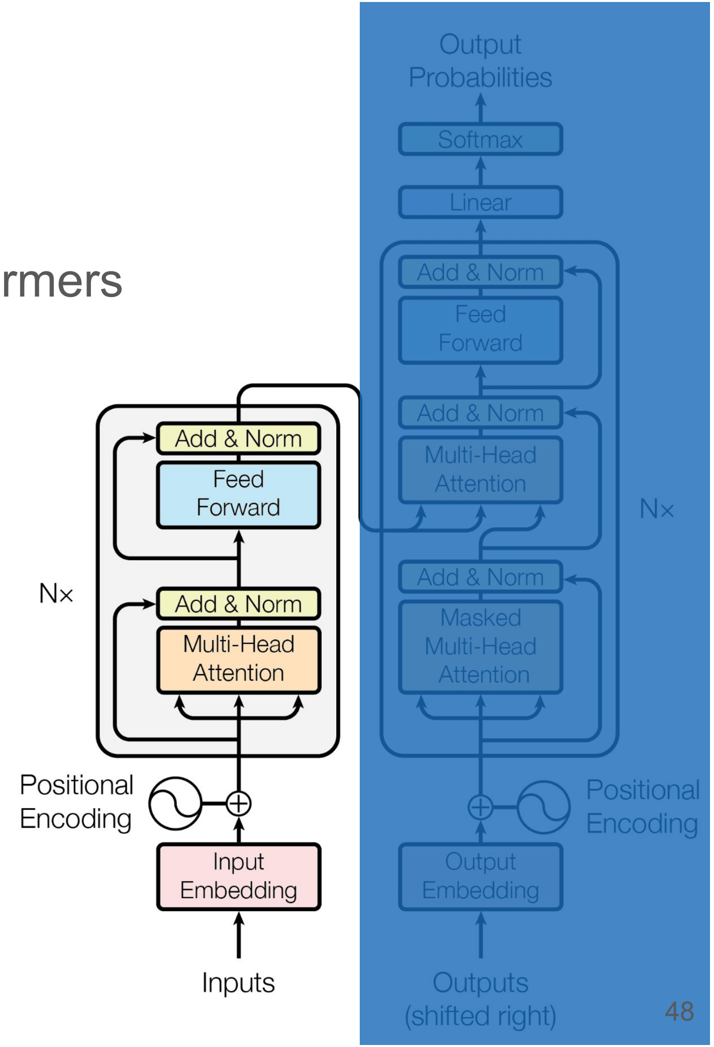
Decoder only



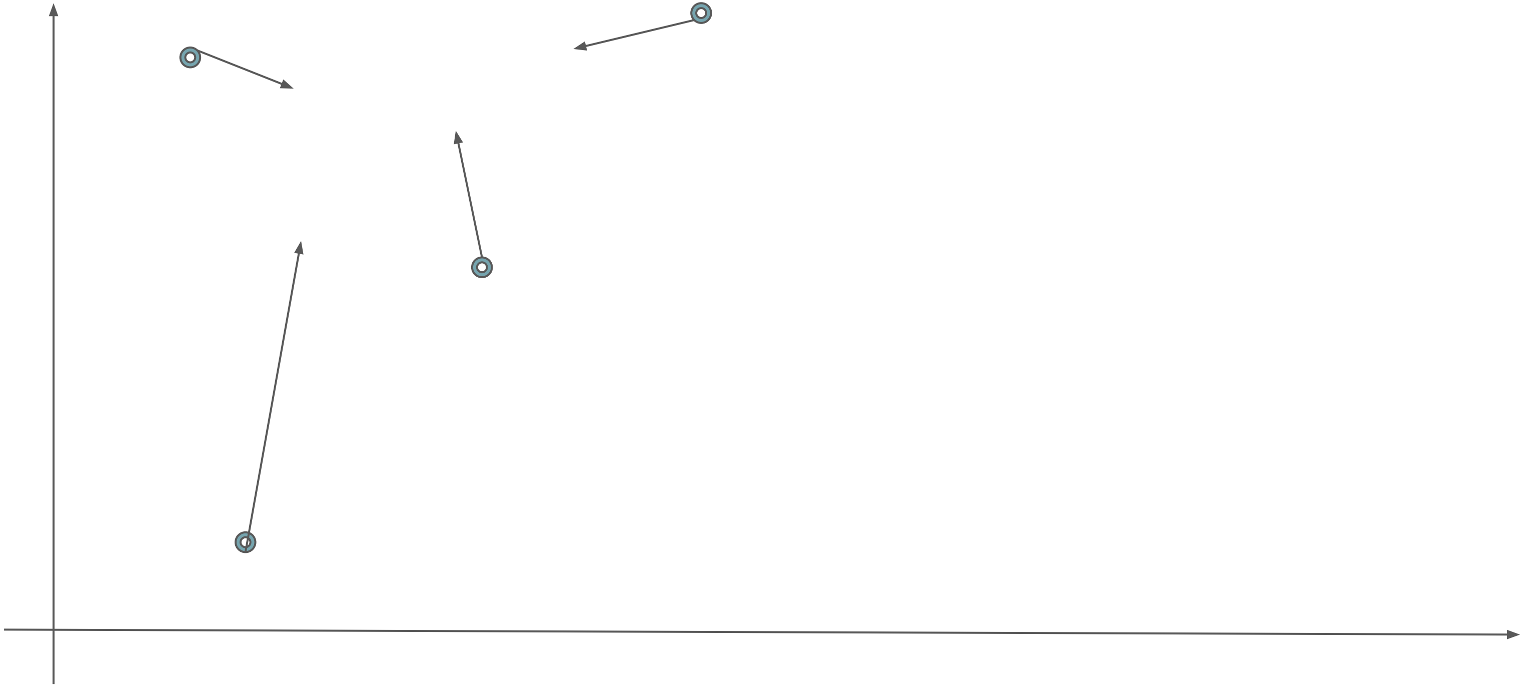
BERT

Bidirectional Encoder Representations from Transformers

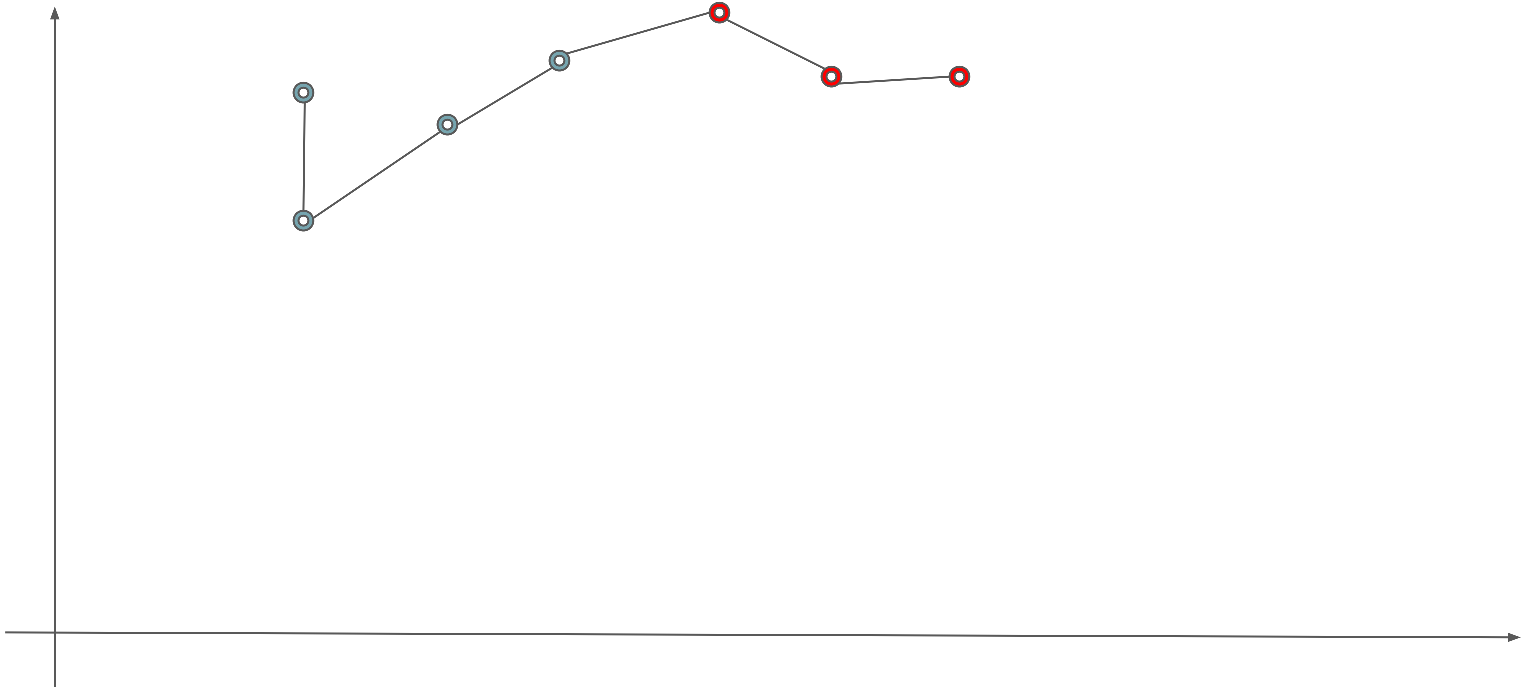
Encoder only



Smart or Stupid

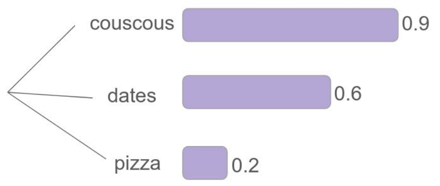


Smart or Stupid

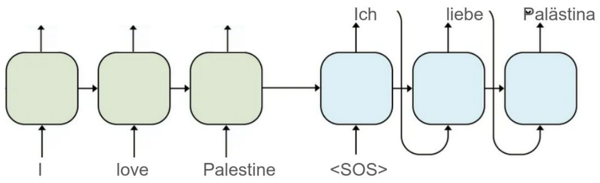


Summary

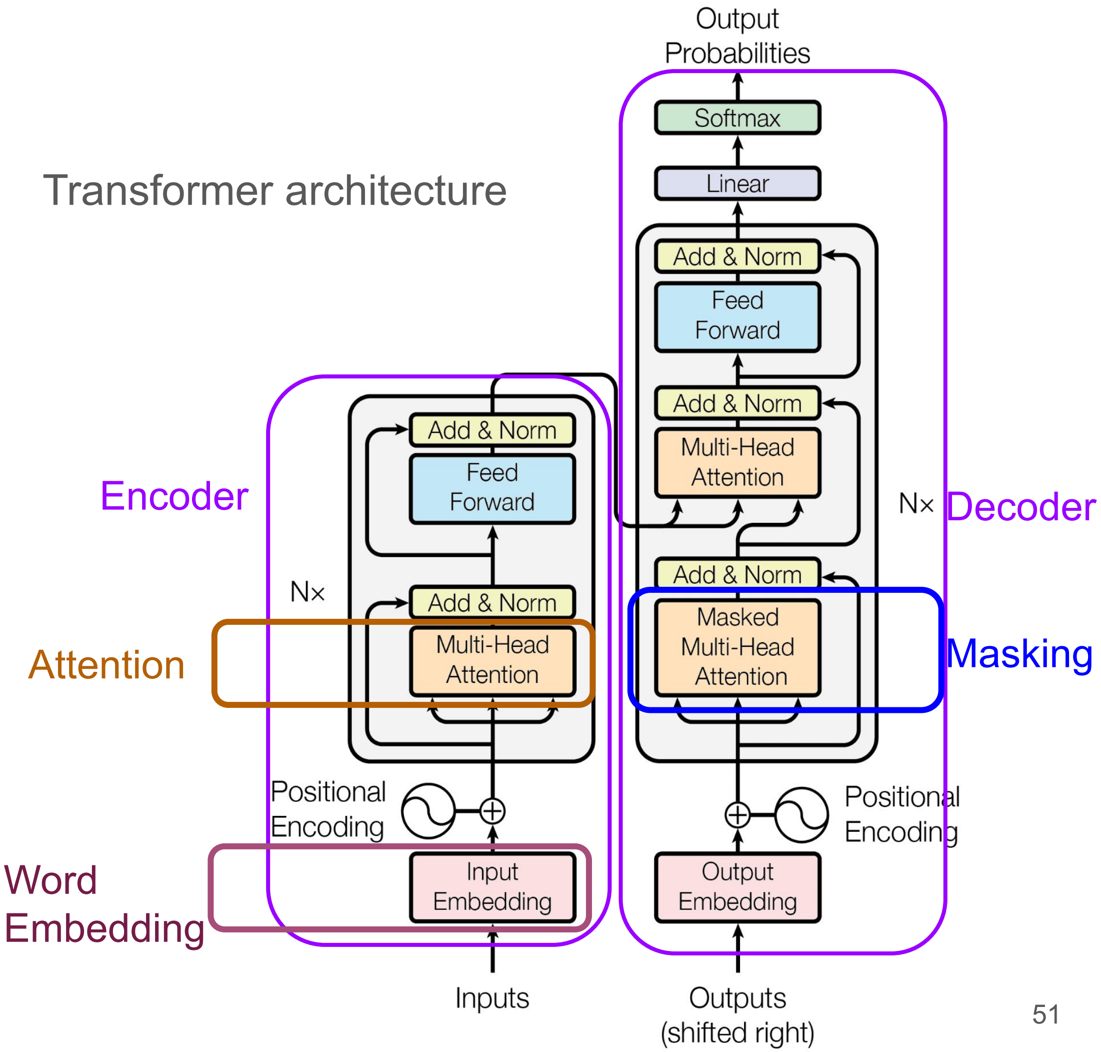
What is a language model



Before the transformer



Transformer architecture



References

Vaswani, Ashish, et al. "*Attention is all you need*." Advances in neural information processing systems 30 (2017).

[1hr Talk] Intro to Large Language Models. Andrej Karpathy
https://youtu.be/zjkBMFhNj_g?si=wpJVQf6ah18LM30z

The Attention Mechanism in Large Language Models. Serrano.Academy
https://youtu.be/OxCpWwDCDFQ?si=qpKI2hgWtgAoEH3n