Advanced Computer Science Course Lecture 3 Advanced Language Models

Tishreen University

Computer and automatic control engineering dept.

Master Program- 2024

1st year

Dr. Ali Mahmoud Mayya

Language Models

Sentiment analysis

Feed Forward Fully-connected Neural Networks

Model

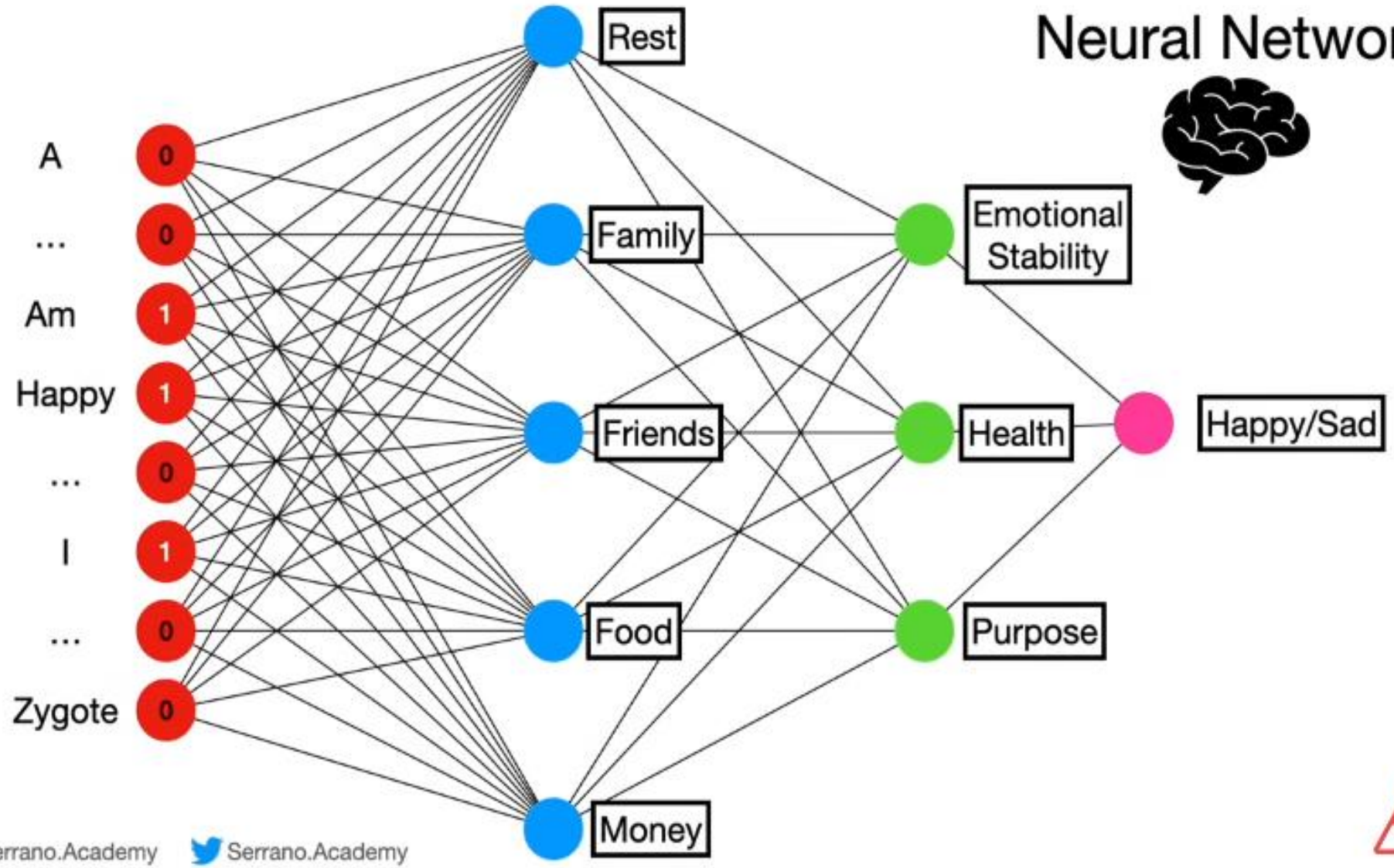
Happy	3 pts.
Sad	-4 pts.
Magnificent	4 pts.
Hello	0 pts.
Very	0 pts.
So	0 pts.
...	...

I am very happy!! 3

0 0 0 3

I had a very bad day -4

0 0 0 0 -4 0



https://www.youtube.com/wat Serrano.Academy Serrano.Academy

Language Models

Sentiment analysis

Predicting next word in the sentence (simple start)

Feed Forward Fully-connected Neural Networks

Model

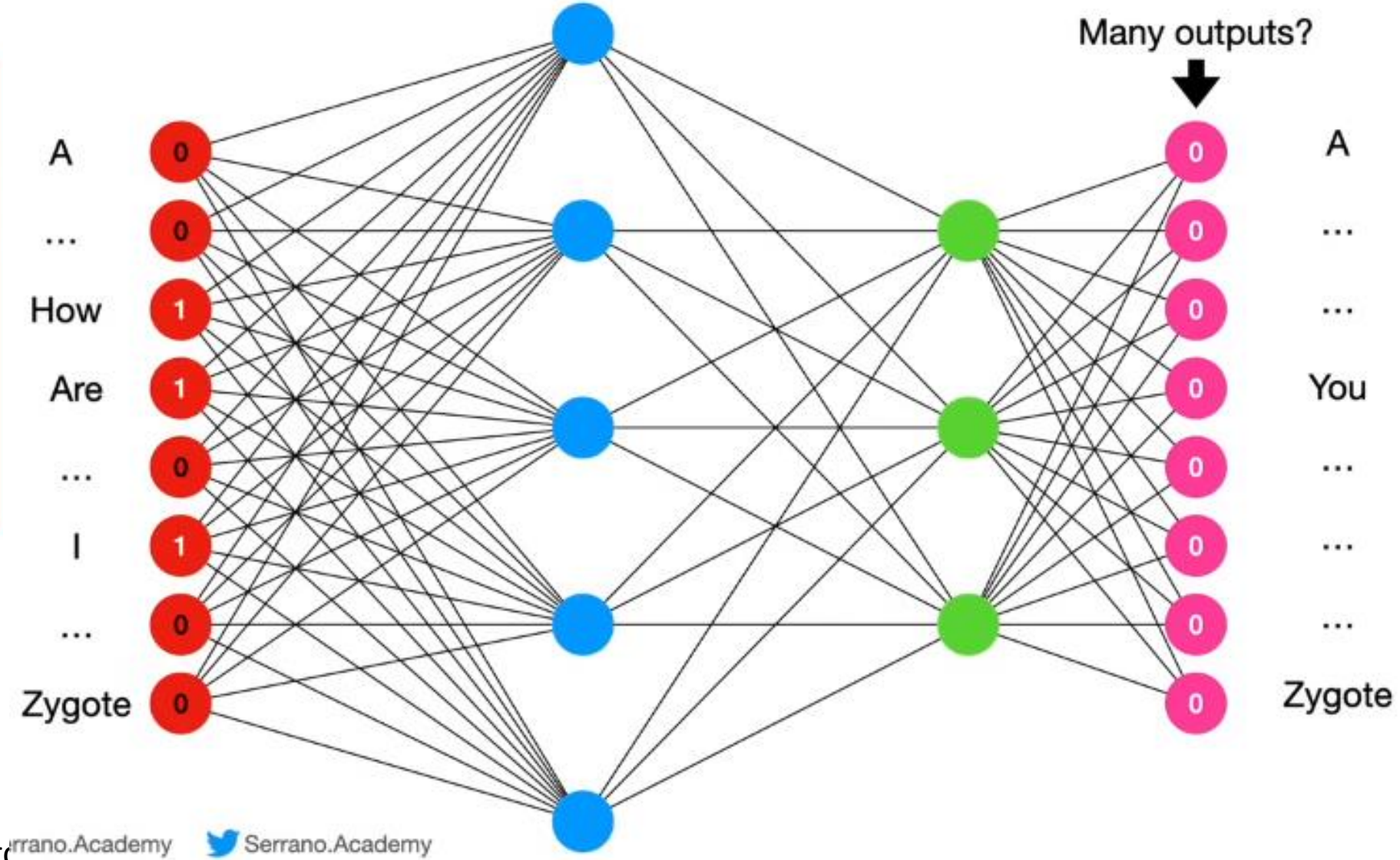
Happy	3 pts.
Sad	-4 pts.
Magnificent	4 pts.
Hello	0 pts.
Very	0 pts.
So	0 pts.
...	...

I am very happy!! 3

0 0 0 3

I had a very bad day -4

0 0 0 0 -4 0



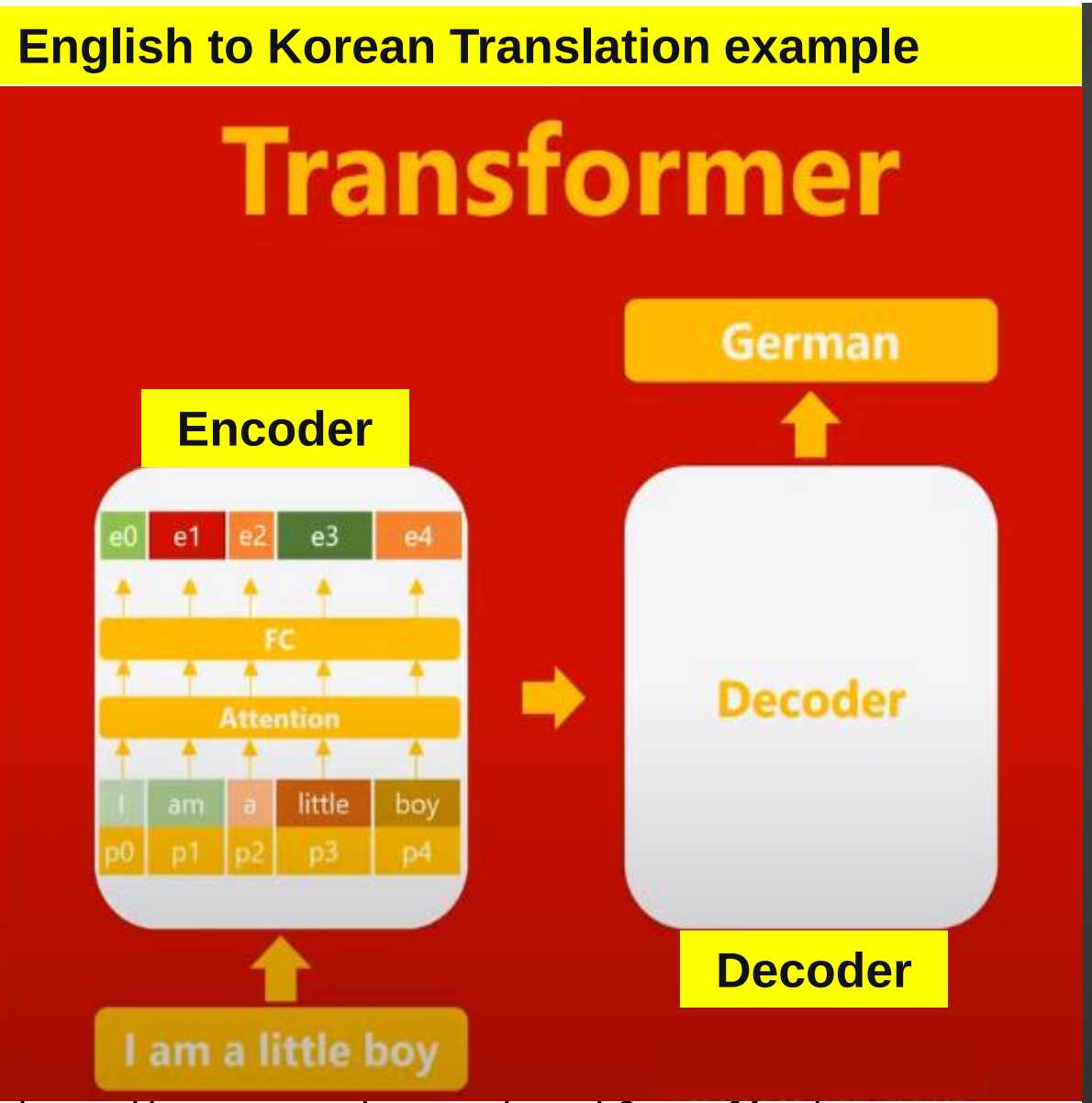
Serrano.Academy

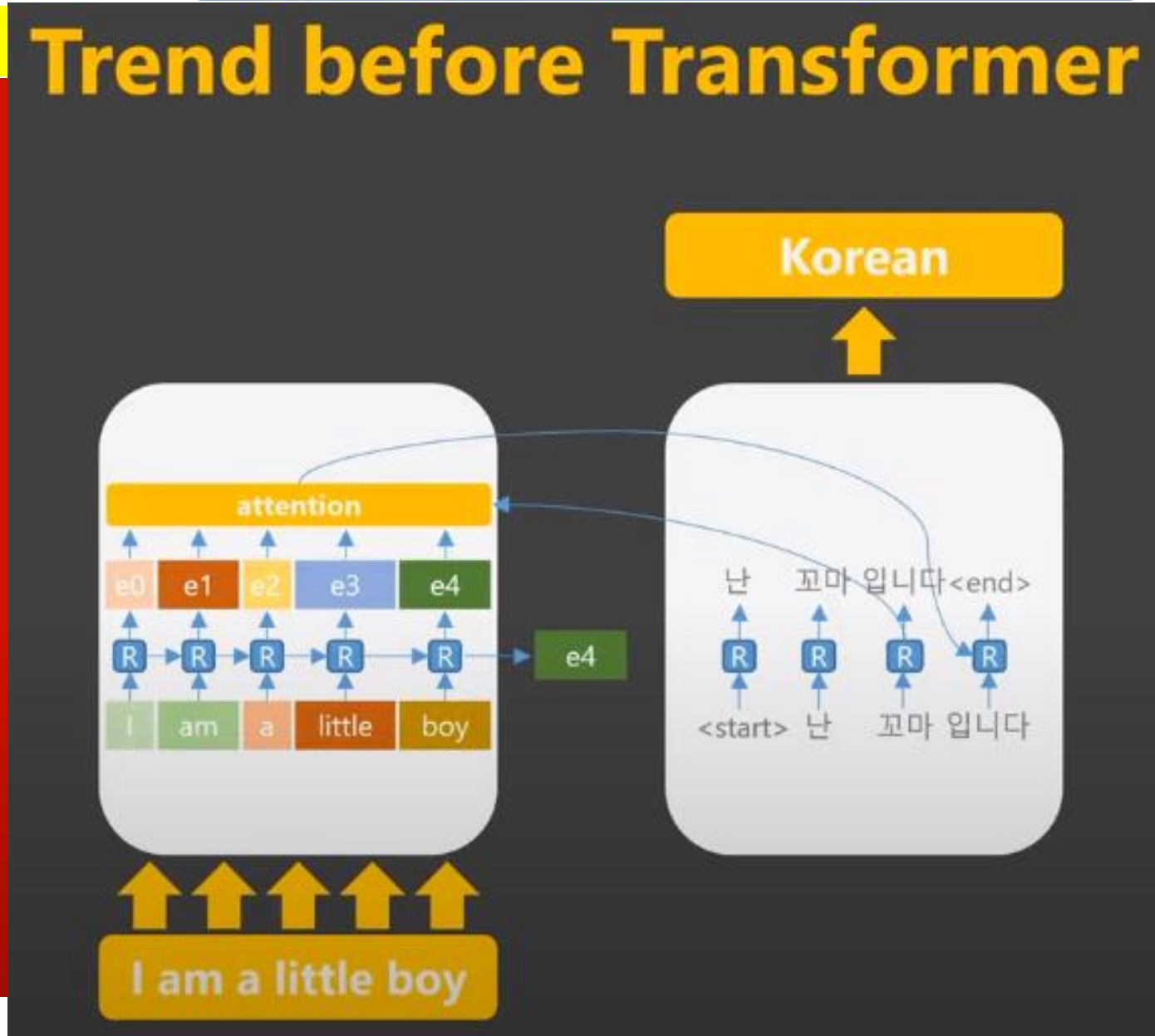
https://www.youtube.com/watch...

Language Models

Transformer models

English to Korean Translation example





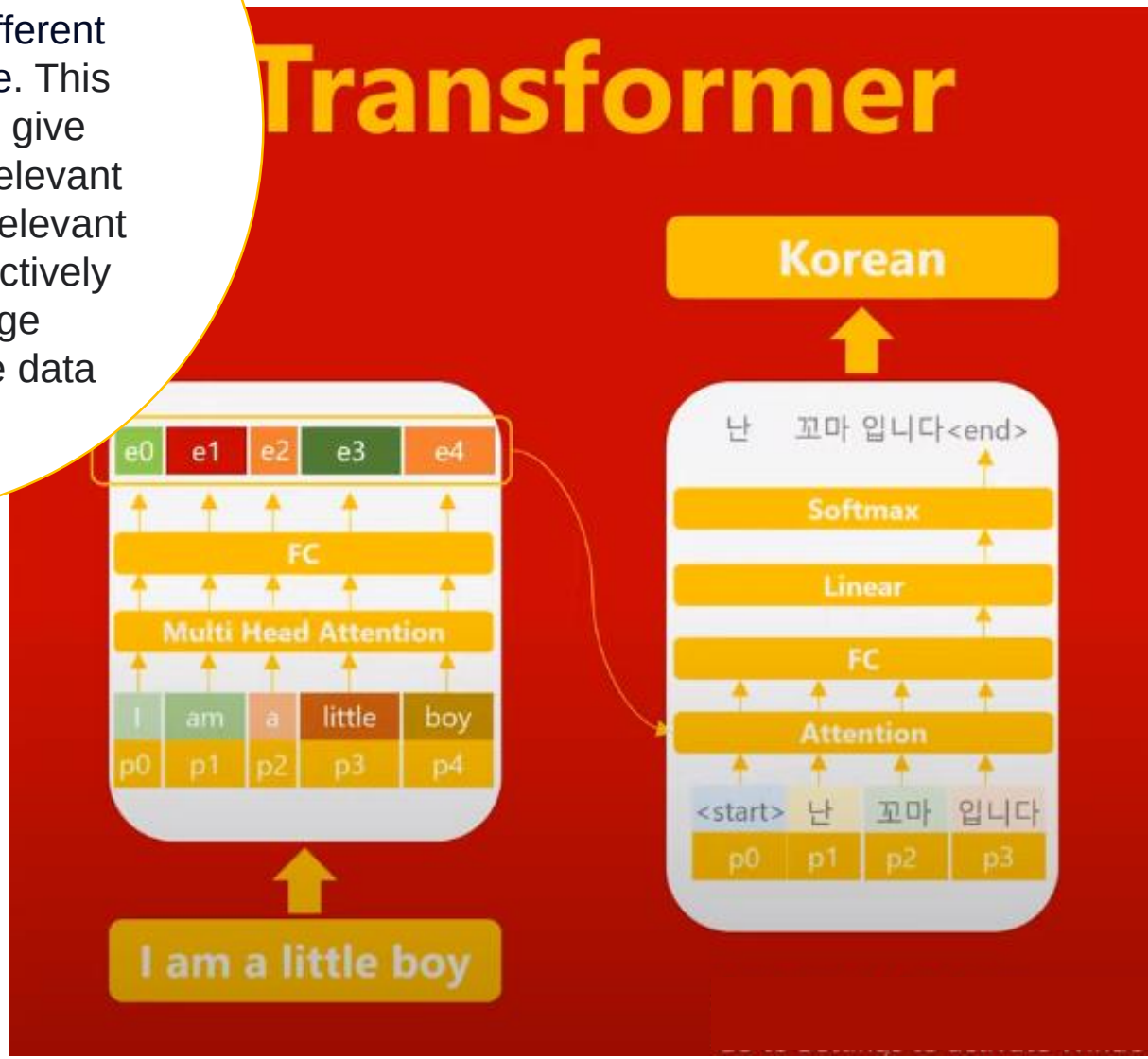
https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

Transformer models

English to Korean Translation ex

the ***attention mechanism*** enables the model to assign different attention weights/scores to different tokens in a sequence. This way, the model can give more importance to relevant information, ignore irrelevant information, and effectively capture long-range dependencies in the data



https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

Transformer models

Encoder part:

Multi-Head attention layers:

1. Computes self-attention scores for each position in the input sequence.
2. Employs multiple attention heads to capture different aspects of context.
3. The output is a weighted sum of values, where the weights are determined by the attention scores.

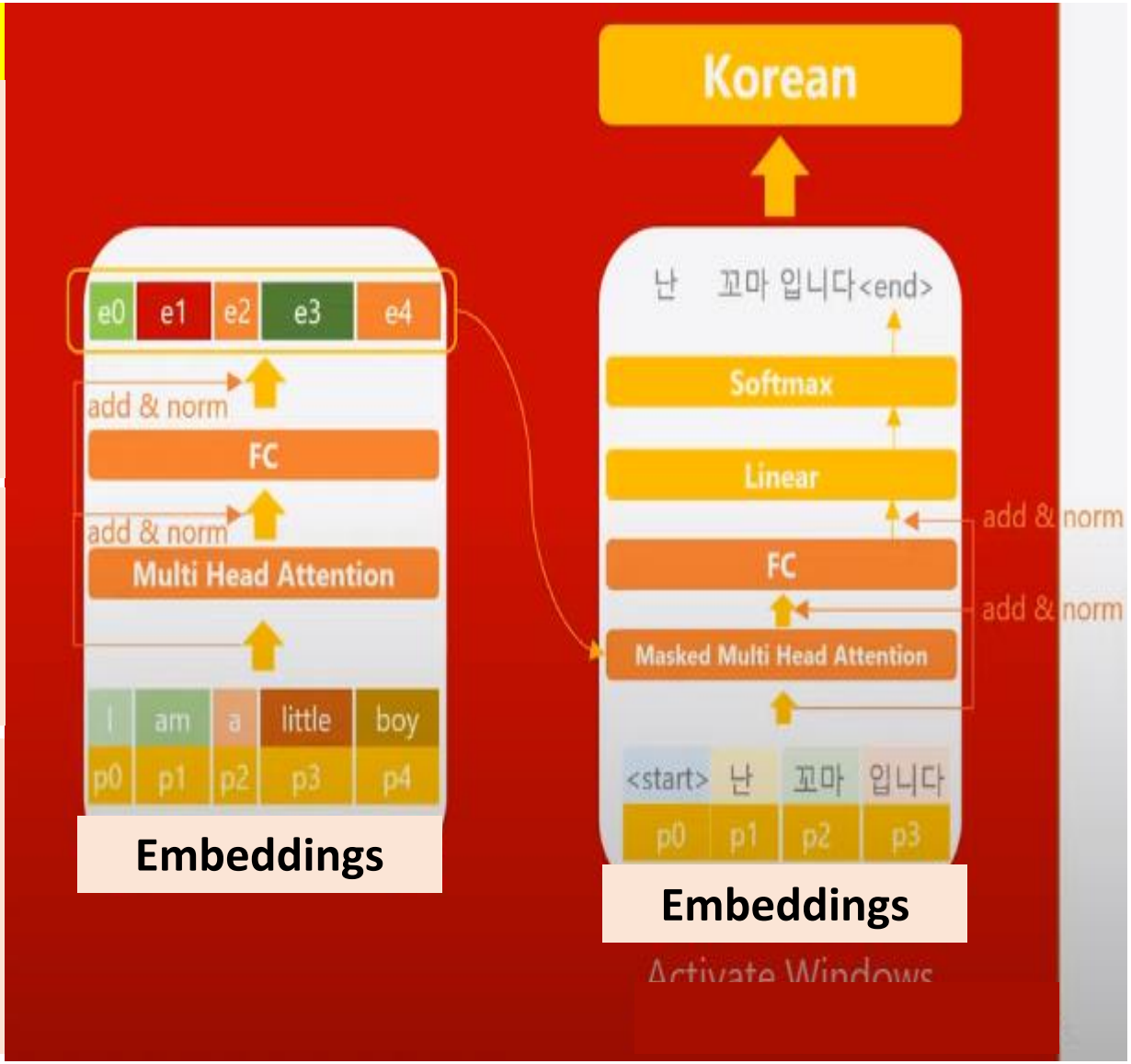
Feed-Forward Network:

1. Fully connected feed-forward network.
2. Two linear transformations followed by a Relu function.
3. Non-linearity and helps model complex relationships.

Add&Norm skip connections:

1- Deliver lower layer's information to the higher layers very efficiently.

2- Enhance training process especially with too deep model (Eliminate vanishing gradient problem)



https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

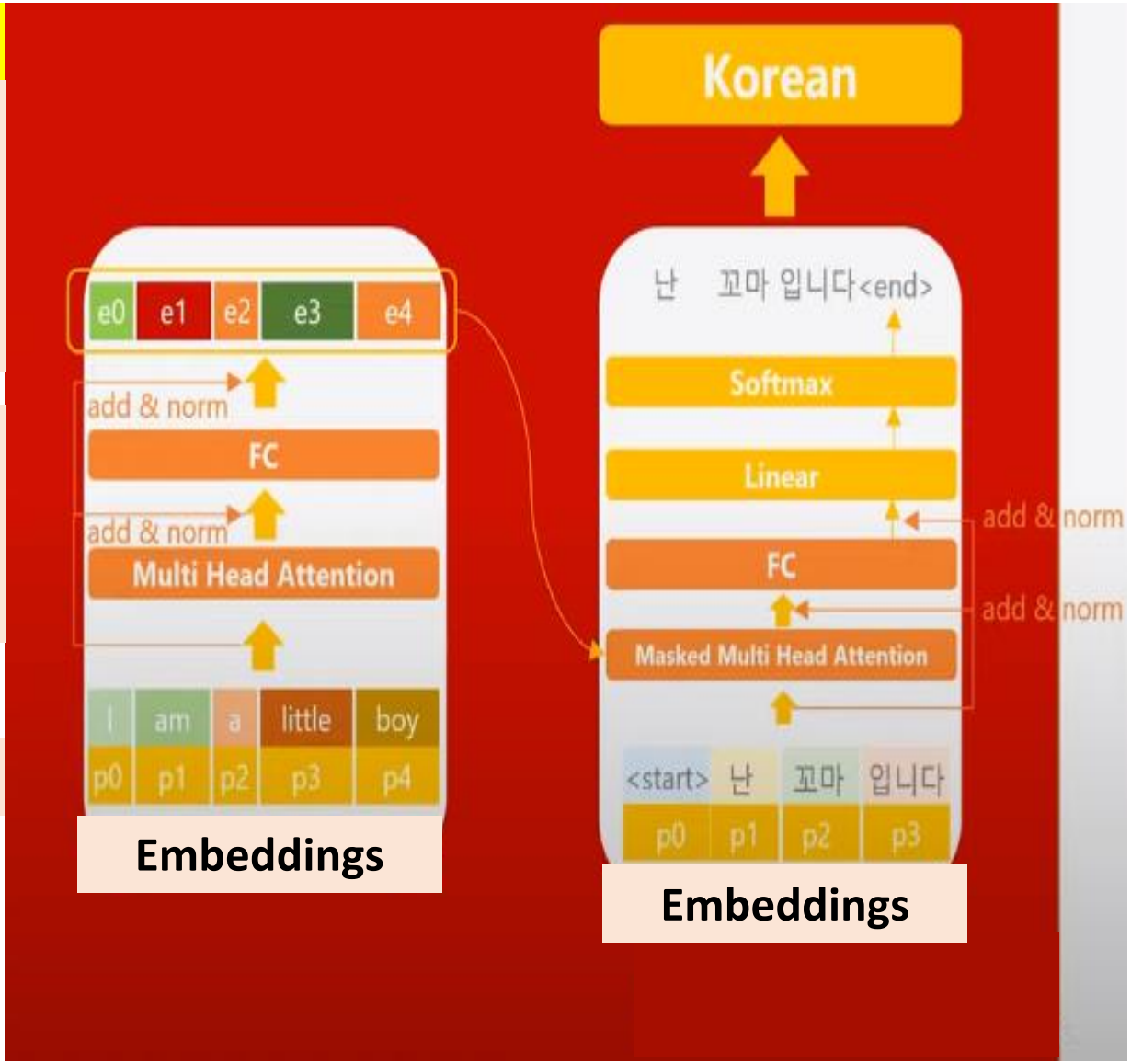
Transformer models

Decoder part:

Masked Multi-Head attention layers:
Like the encoder's self-attention, but with a mask to prevent attending to future positions.
Ensures that the decoder only uses information from previous positions during prediction.

Feed-Forward Network:
1. Fully connected feed-forward network.
2. Two linear transformations followed by a Relu function.
3. Helps model complex relationships.

Linear layer with softmax activation for classification



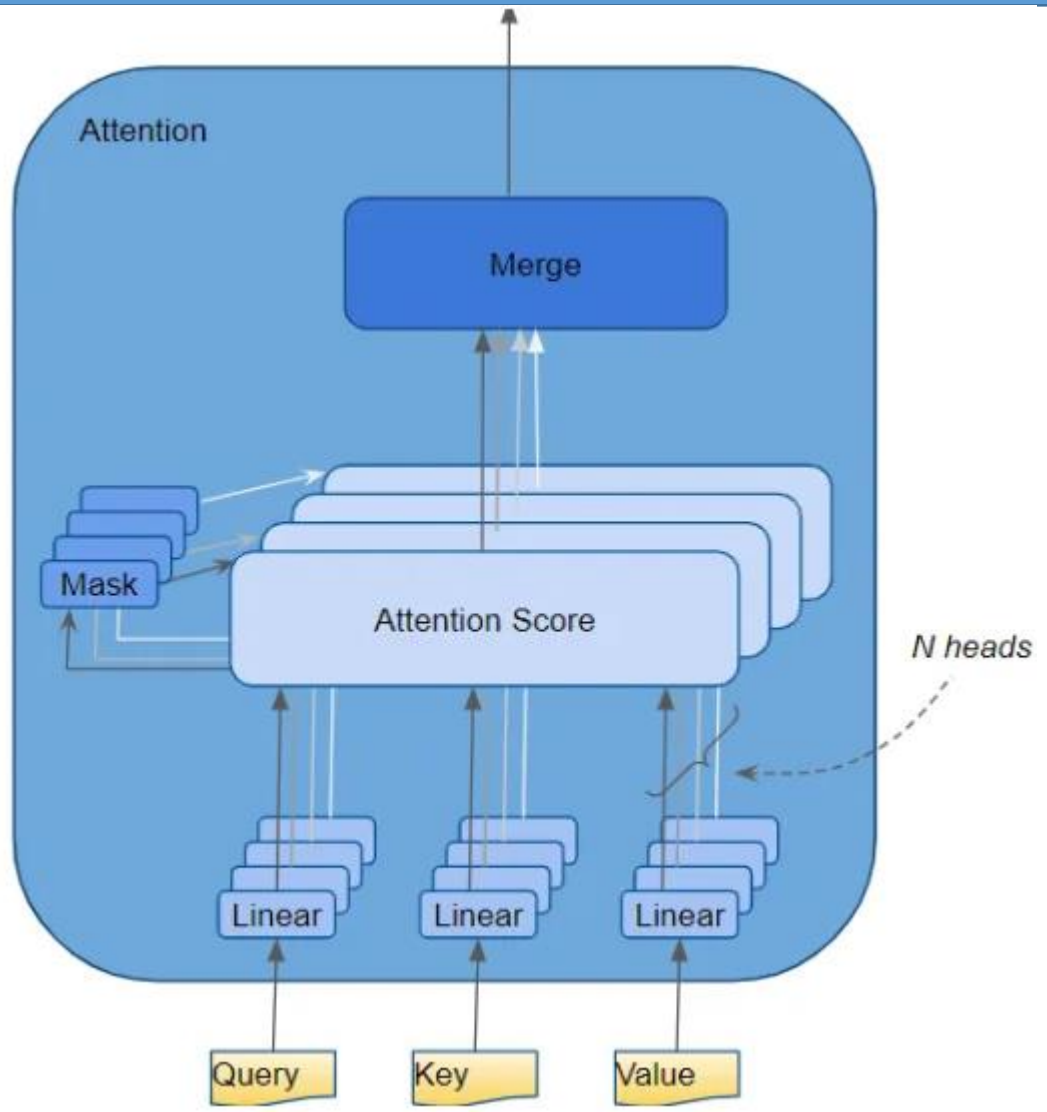
https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

Transformer models

Multi-head attention layer architecture

- The Attention module splits its Query, Key, and Value parameters N-ways
- Passes each split independently through a separate Head.
- Attention calculations are then combined together to produce a final Attention score.

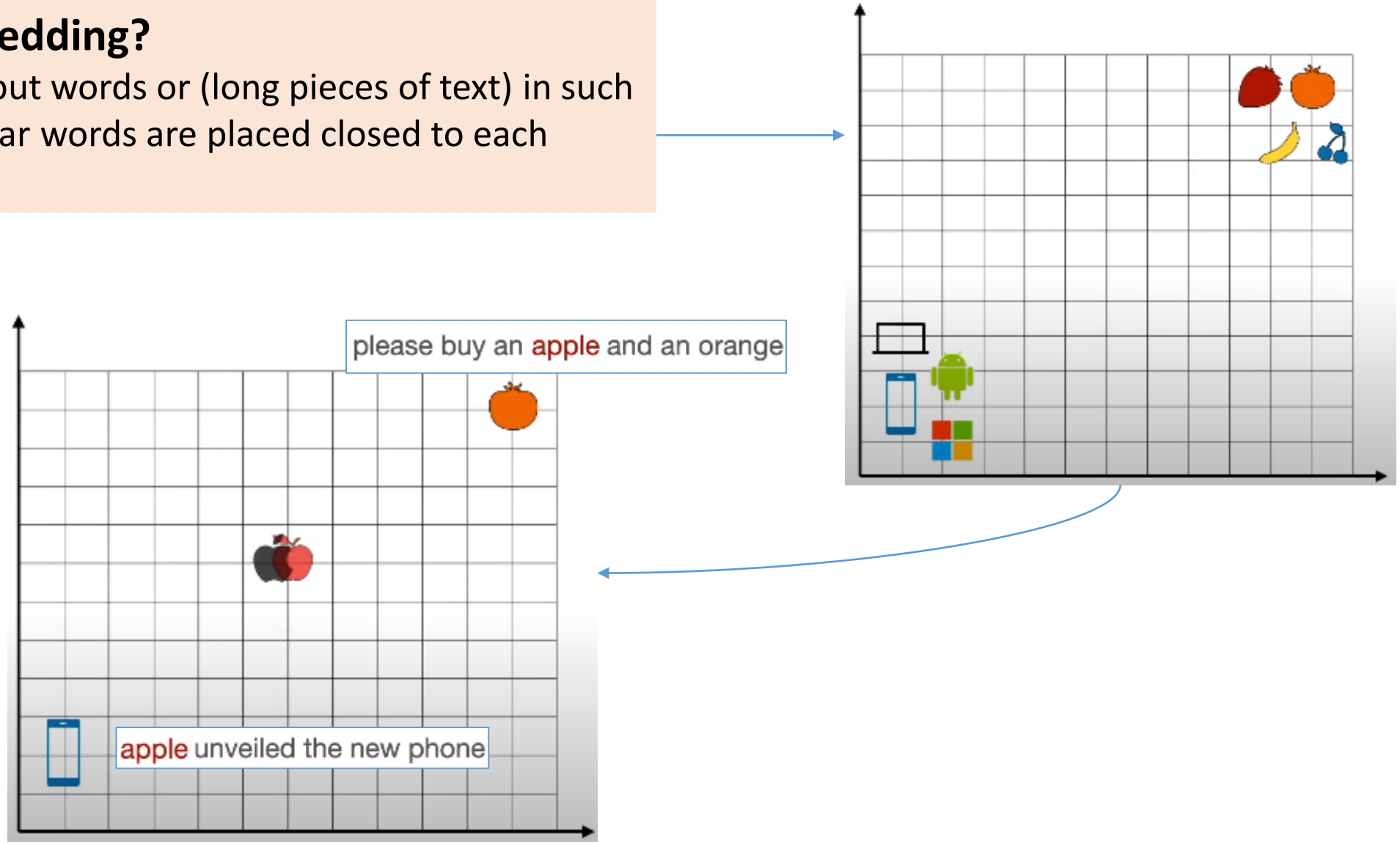


Language Models

Transformer models

What is embedding?

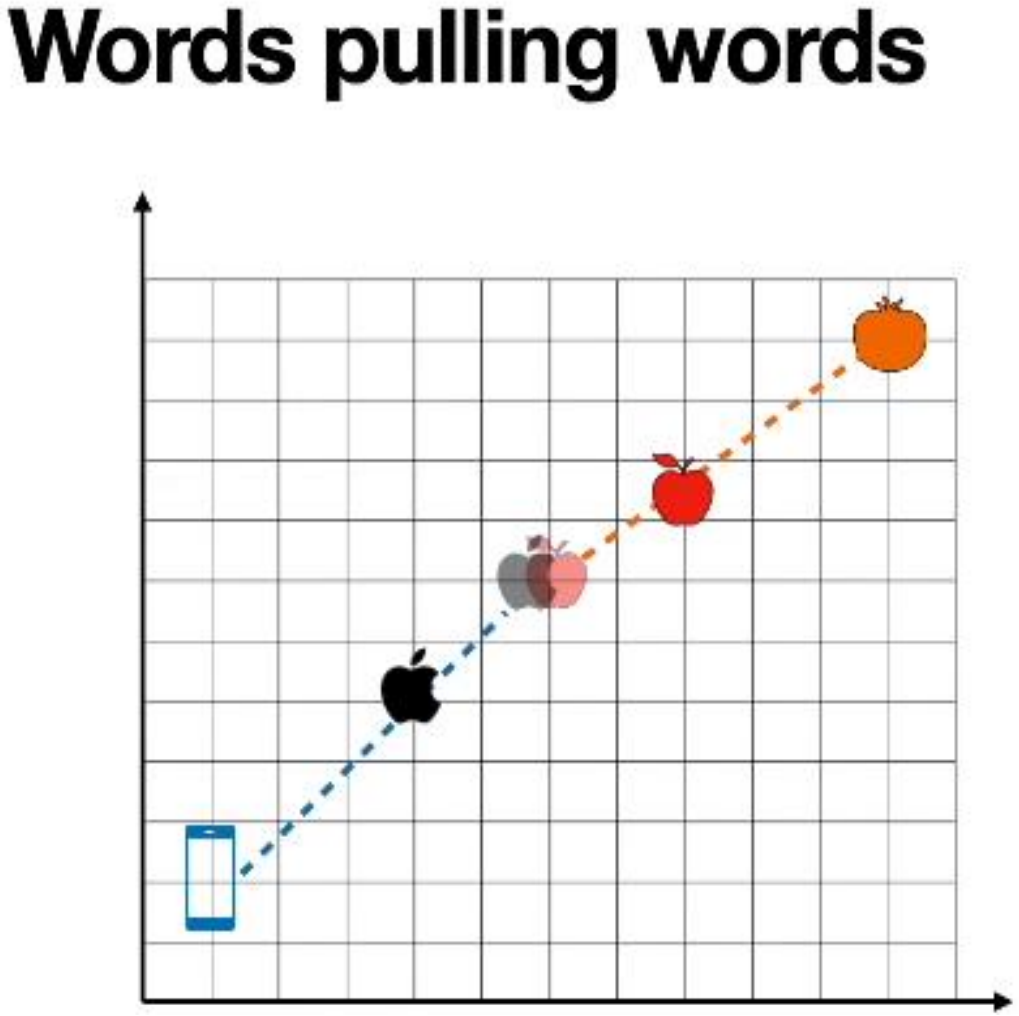
Mechanism to put words or (long pieces of text) in such a way that similar words are placed closed to each others.



https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models



https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

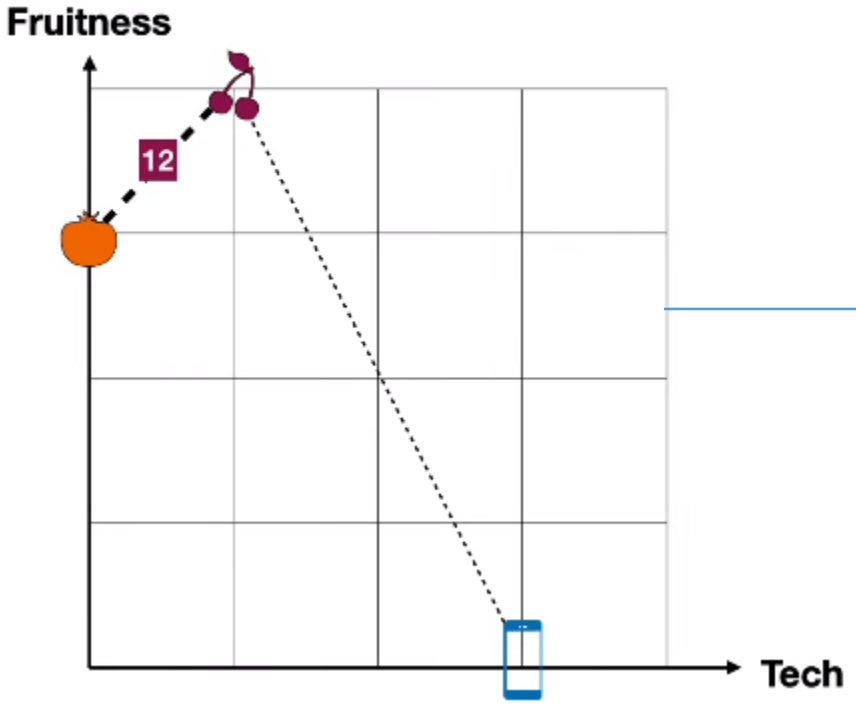
Transformer models

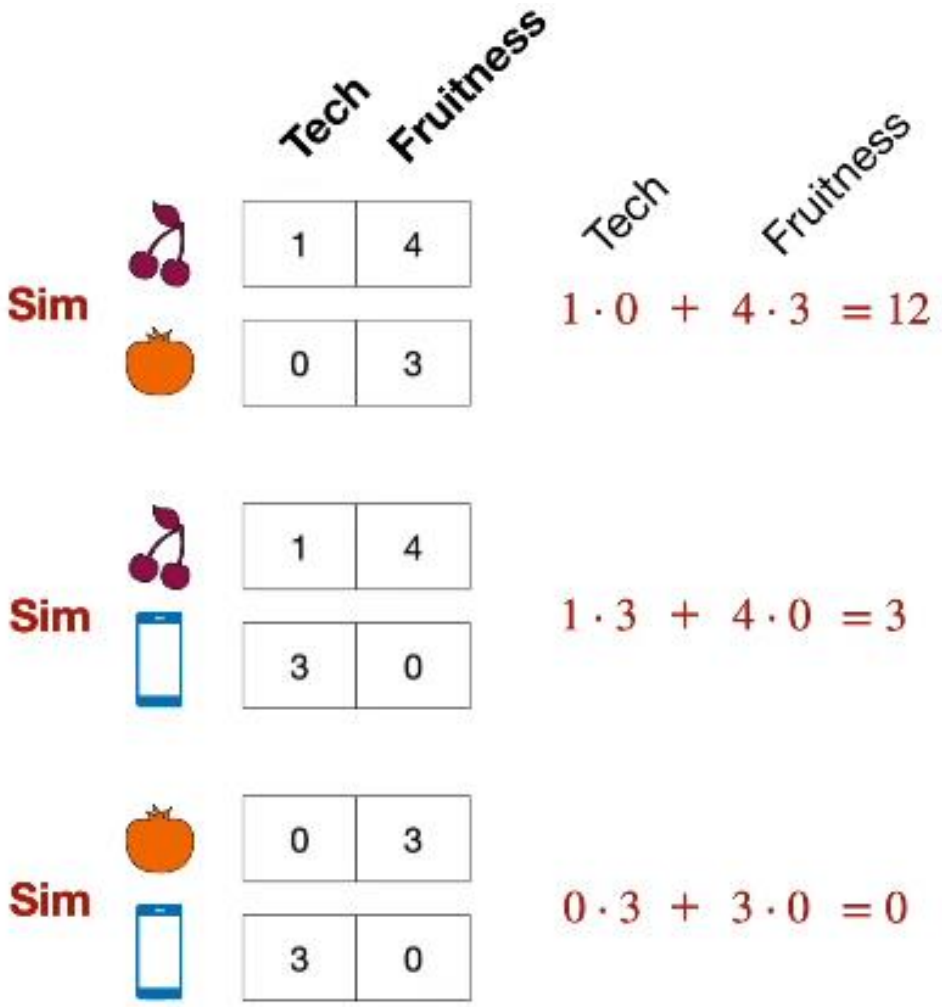
Language Models

Transformer models

Similarity

Measure 1: Dot product



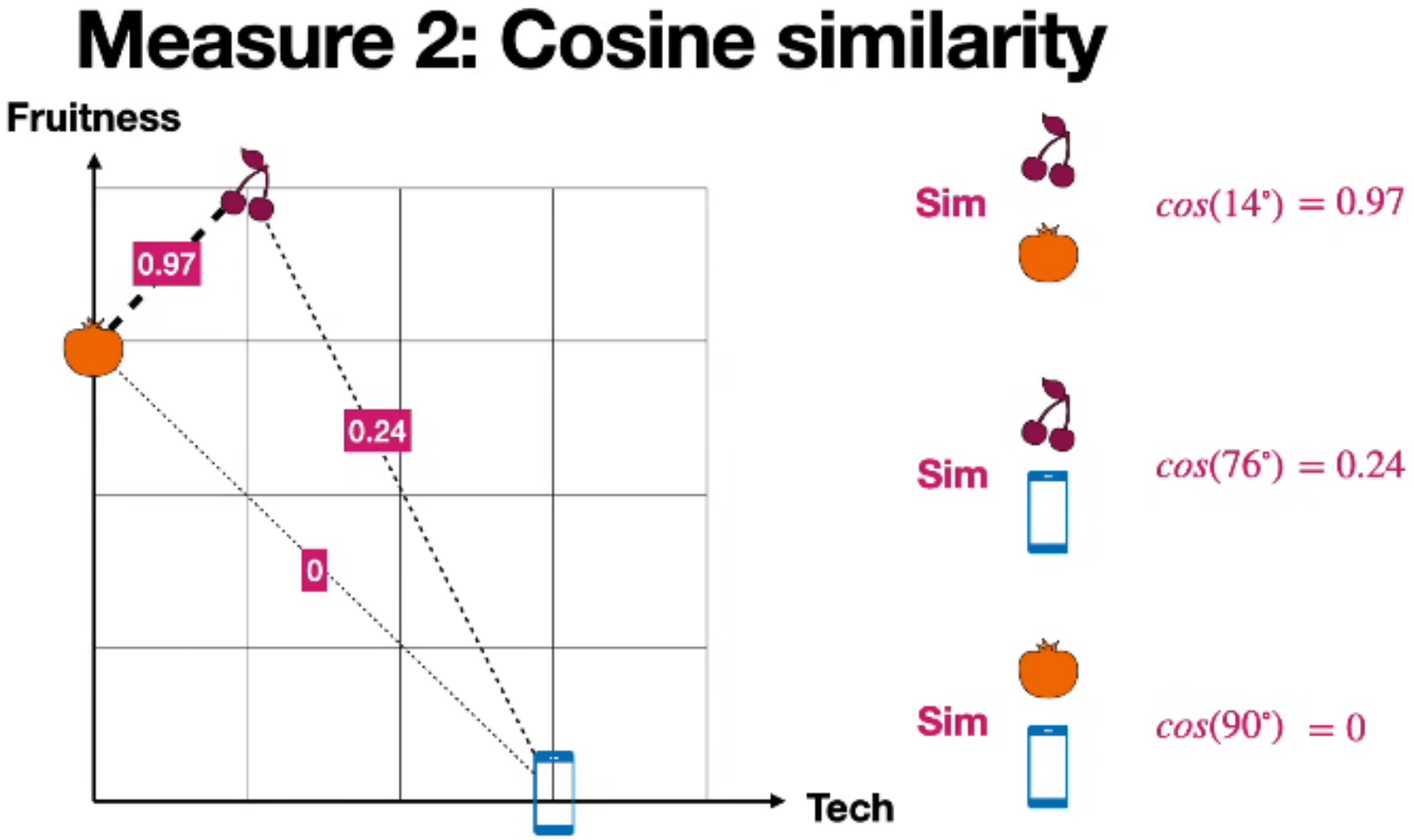


https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



https://www.youtube.com/watch?v=UPtG_38Oq8o

Similarity

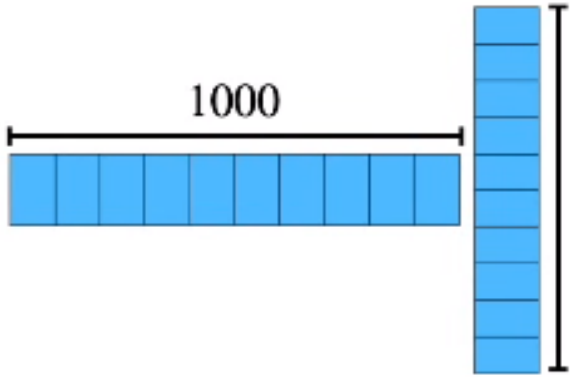
Measure 3: Scaled dot product

Dot product divided by the square root of the length of the vector

Sim

1	4
0	3

$$1 \cdot 0 + 4 \cdot 3 = 12 \longrightarrow \frac{12}{\sqrt{2}} = 8.49$$



Sim

1	4
3	0

$$1 \cdot 3 + 4 \cdot 0 = 3 \longrightarrow \frac{3}{\sqrt{2}} = 2.12$$

Sim

0	3
3	0

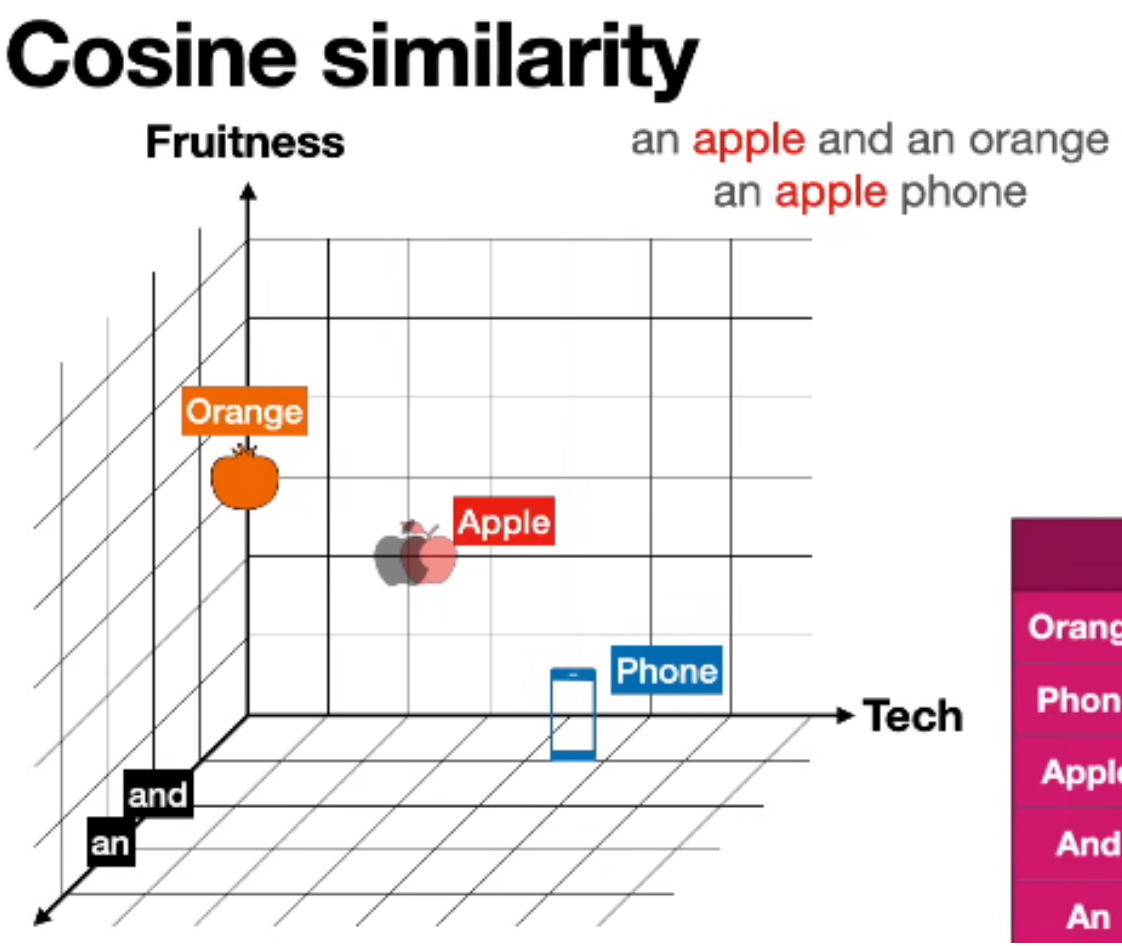
$$0 \cdot 3 + 3 \cdot 0 = 0 \longrightarrow \frac{0}{\sqrt{2}} = 0$$

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

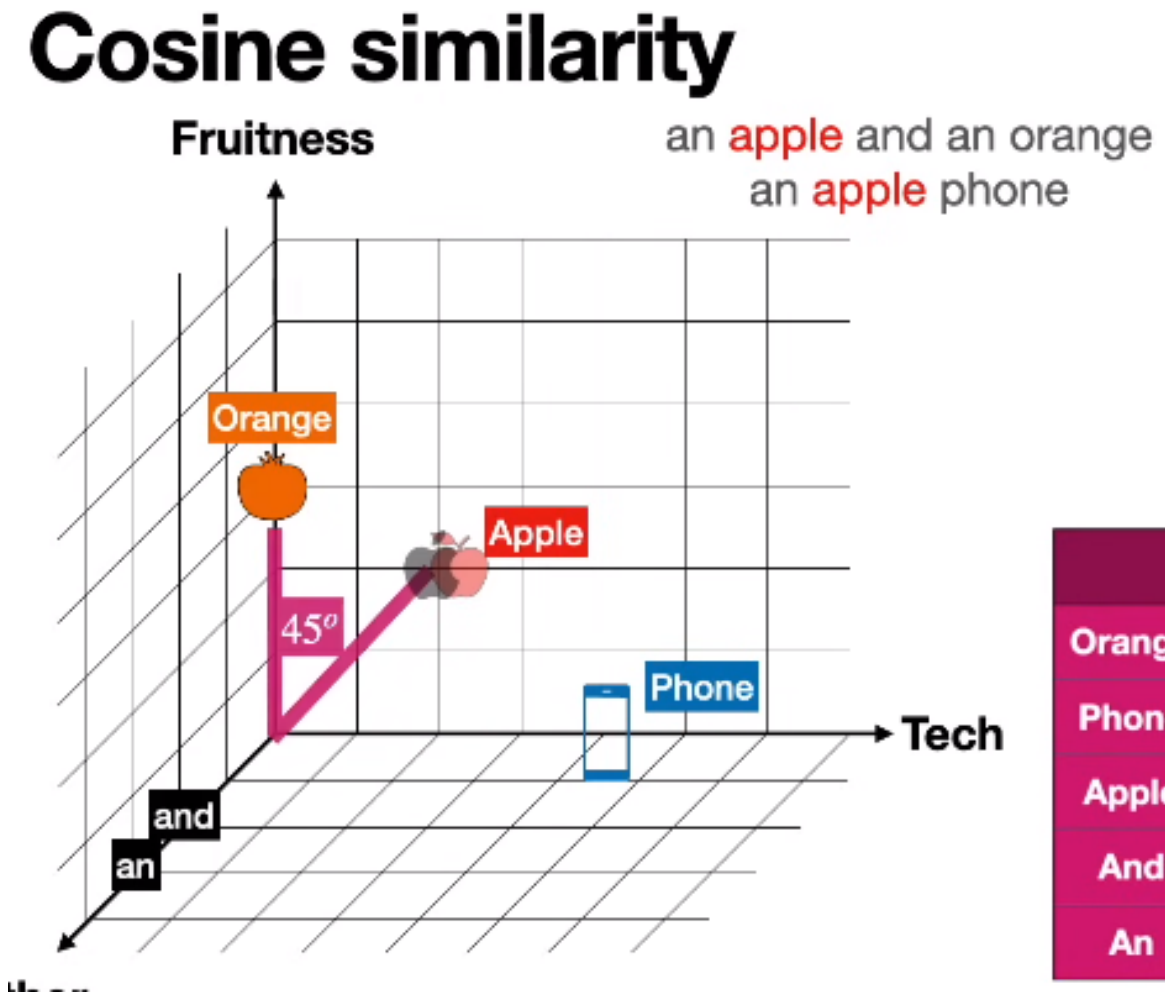
	Orange	Phone	Apple	And	An
Orange	1				
Phone		1			
Apple			1		
And				1	
An					1

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

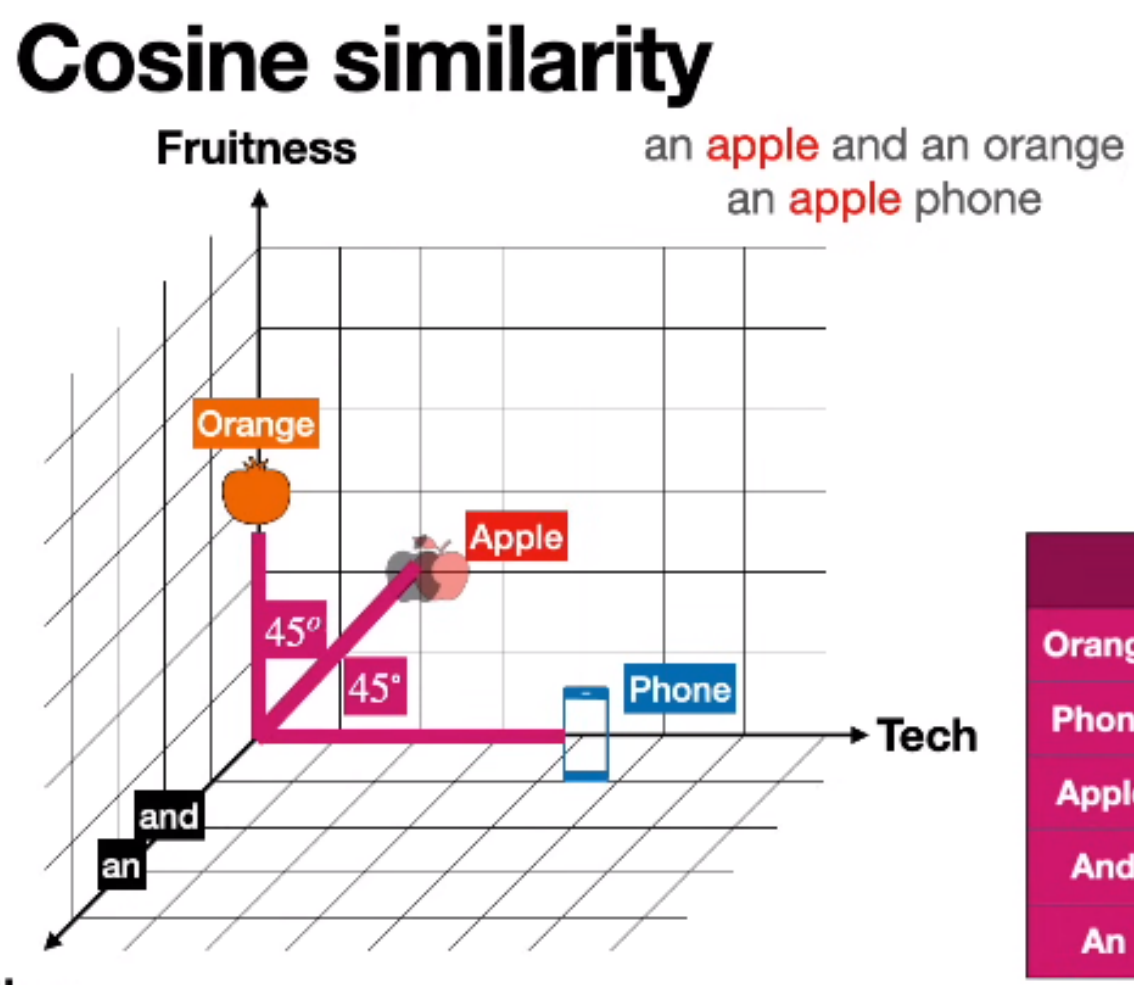
	Orange	Phone	Apple	And	An
Orange	1	0			
Phone	0	1			
Apple			1		
And				1	
An					1

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

	Orange	Phone	Apple	And	An
Orange	1	0	0.71		
Phone	0	1	0.71		
Apple	0.71	0.71	1		
And				1	
An					1

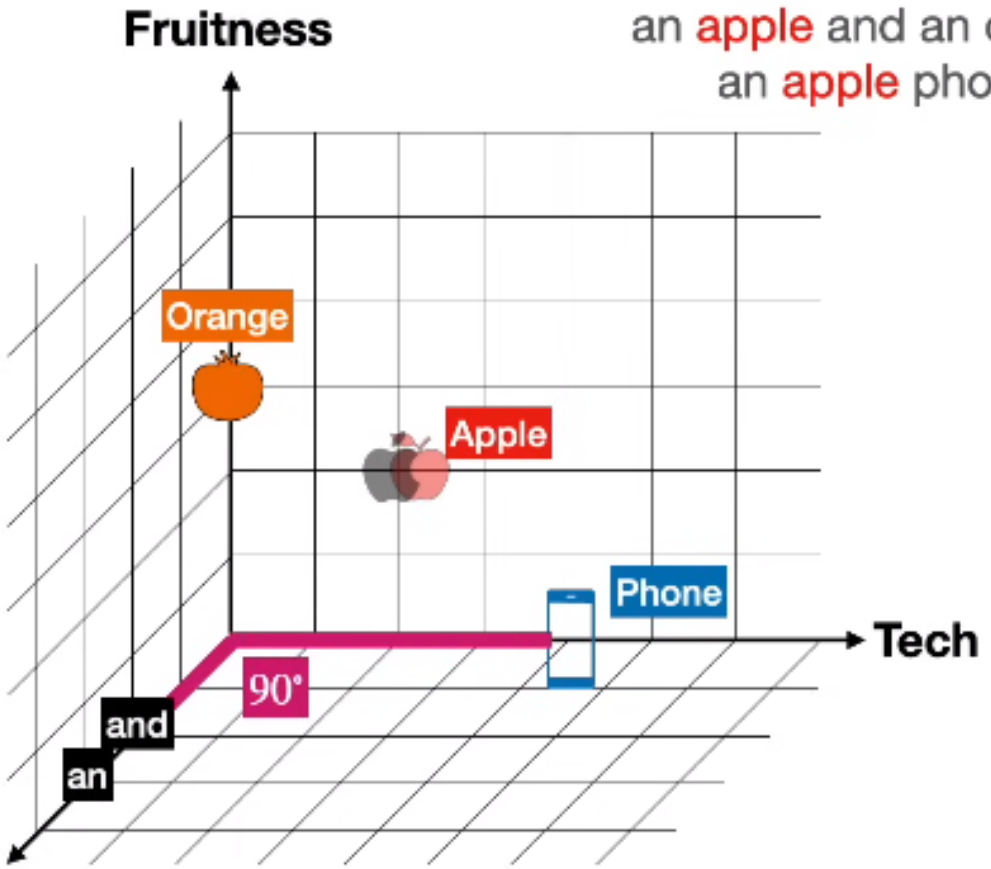
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity

Cosine similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

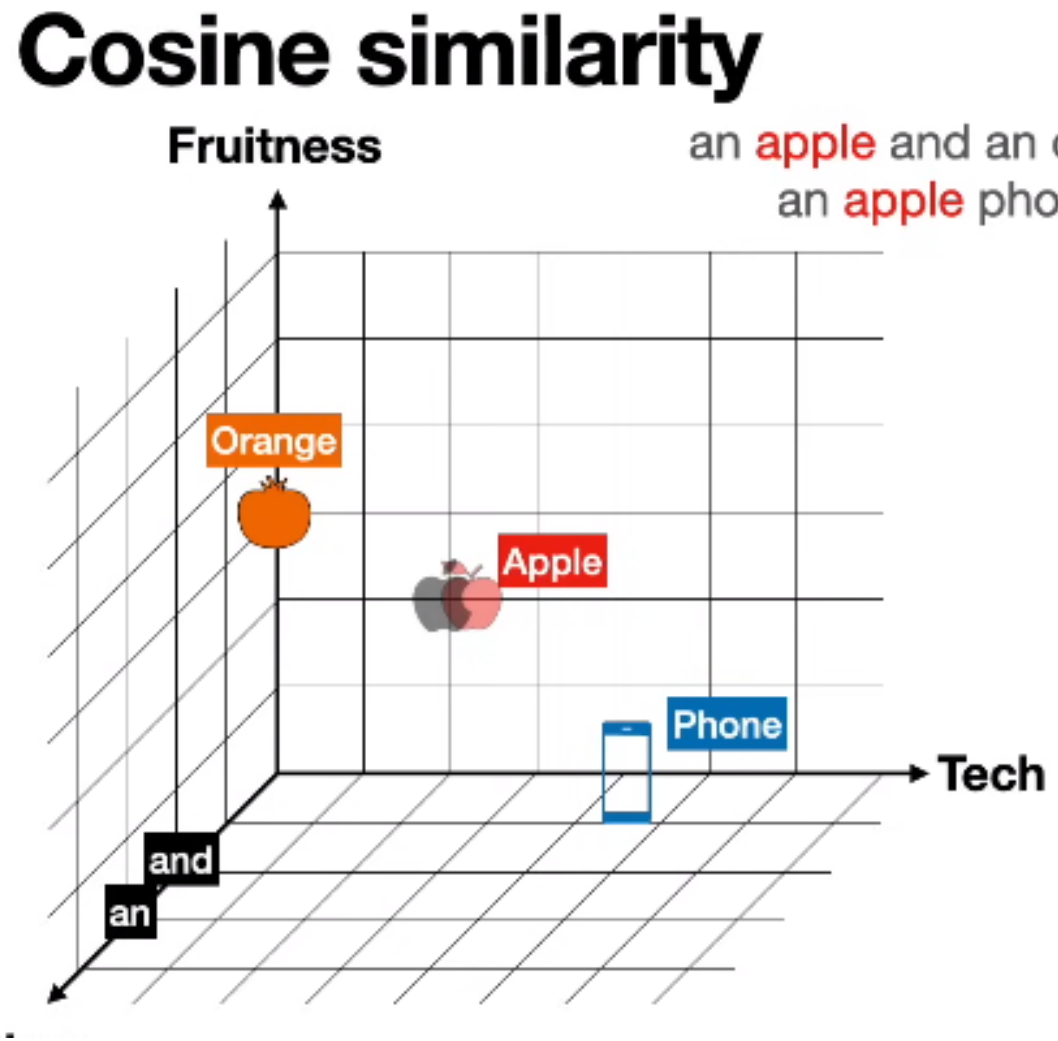
	Orange	Phone	Apple	And	An
Orange	1	0	0.71		
Phone	0	1	0.71		
Apple	0.71	0.71	1		
And				1	
An					1

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

	Orange	Phone	Apple	And	An
Orange	1	0	0.71	0	0
Phone	0	1	0.71	0	0
Apple	0.71	0.71	1	0	0
And	0	0	0	1	1
An	0	0	0	1	1

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity

Word math

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Apple $\longrightarrow$ 0.71 Orange + 1 Apple

And $\longrightarrow$ 1 And + 1 An

An $\longrightarrow$ 1 An + 1 And

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity

an apple phone

	Phone	Apple	An
Phone	1	0.71	0
Apple	0.71	1	0
An	0	0	1

Phone → 1 Phone + 0.71 Apple

Apple → 0.71 Phone + 1 Apple

An → 1 An

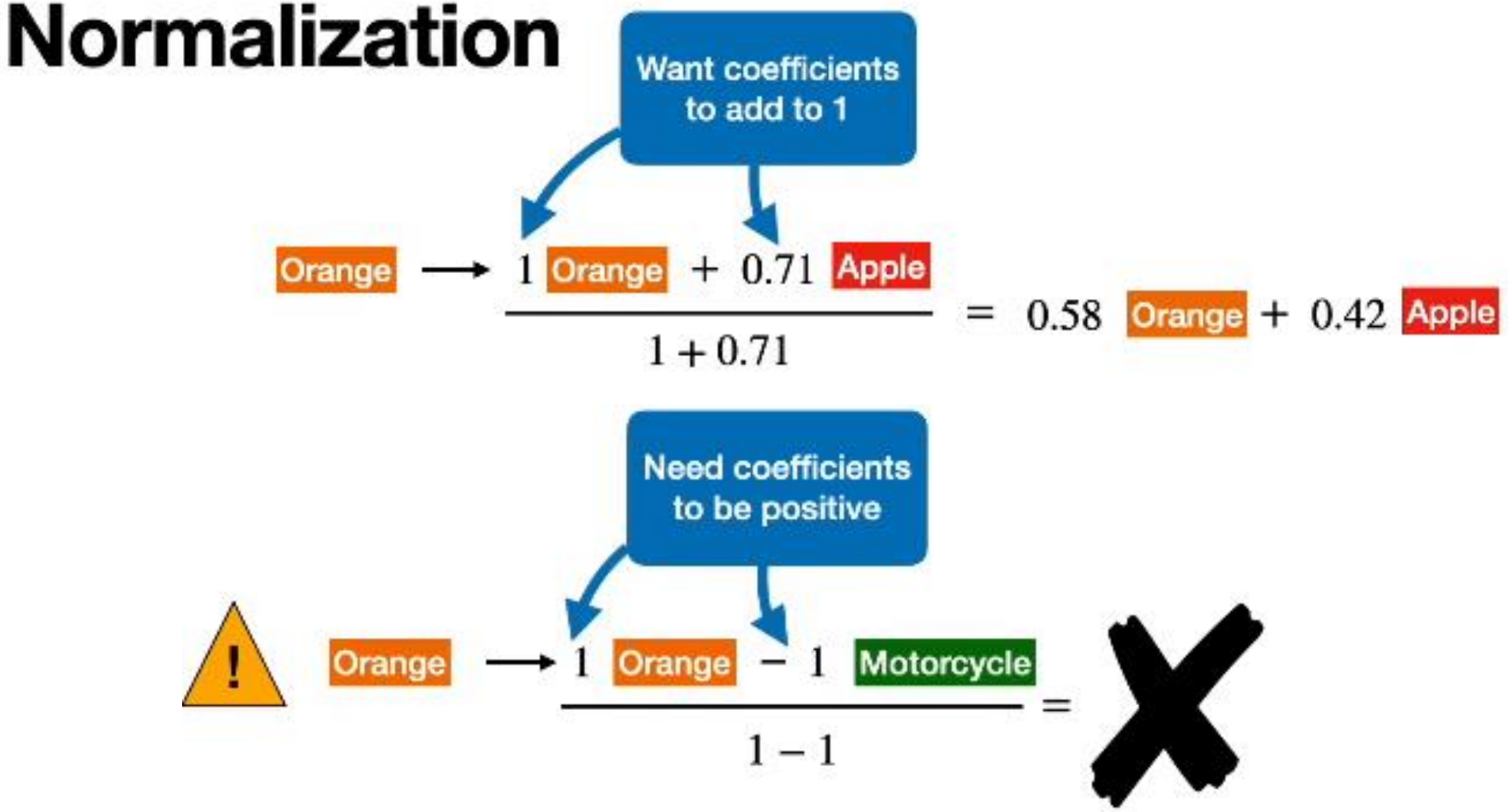
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity

Normalization



https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity

Softmax

$$x \longrightarrow e^{x}$$

$$\text{Orange} \longrightarrow \frac{e^{1}\,\text{Orange} + e^{0.71}\,\text{Apple}}{e^{1} + e^{0.71}} = 0.57\,\text{Orange} + 0.43\,\text{Apple}$$

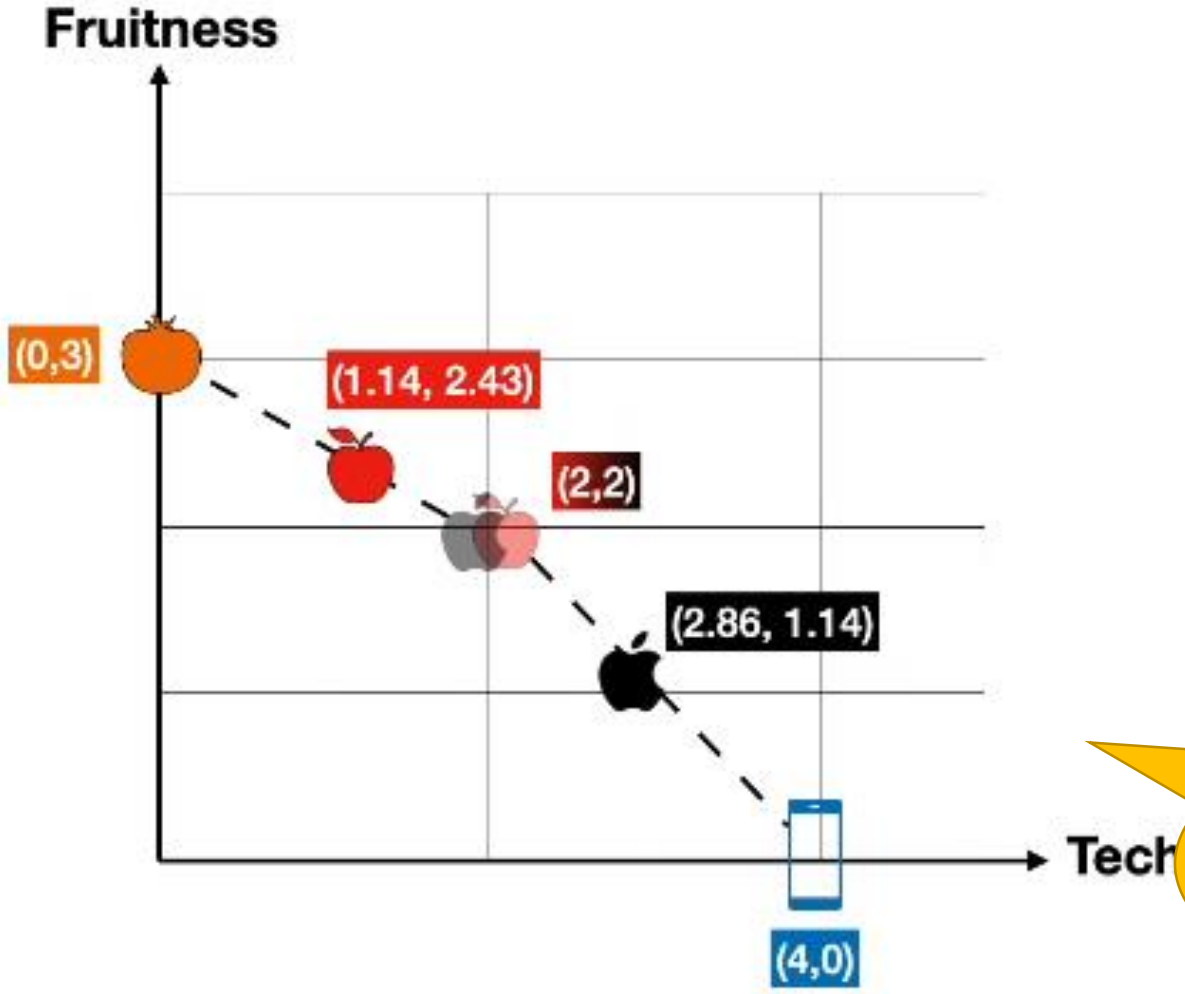
$$\text{Orange} \longrightarrow \frac{e^{1}\,\text{Orange} + e^{-1}\,\text{Motorcycle}}{e^{1} + e^{-1}} = 0.88\,\text{Orange} + 0.12\,\text{Motorcycle}$$

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

Similarity



an apple and an orange

Apple → 0.43 Orange + 0.57 Apple

an apple phone

Apple → 0.43 Phone + 0.57 Apple

Two different coordinates of the same word with different context

In transformer model, exactly with attention mechanism, this operation is performed many times

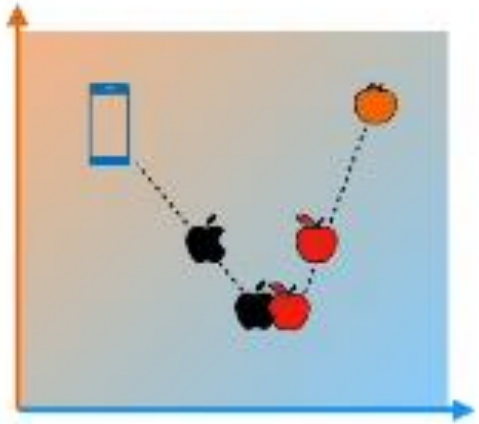
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

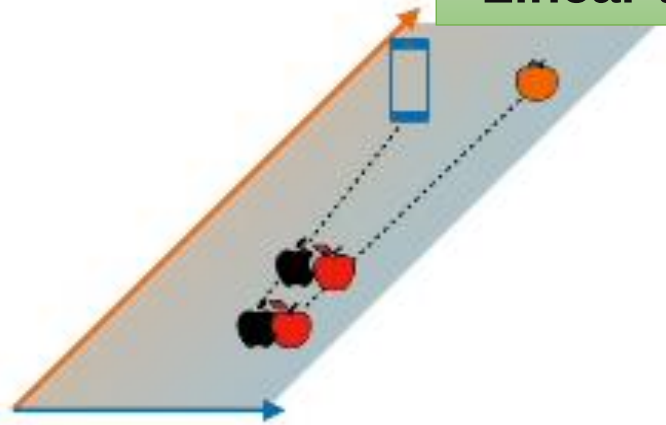
Transformer models

Keys, Query and Values

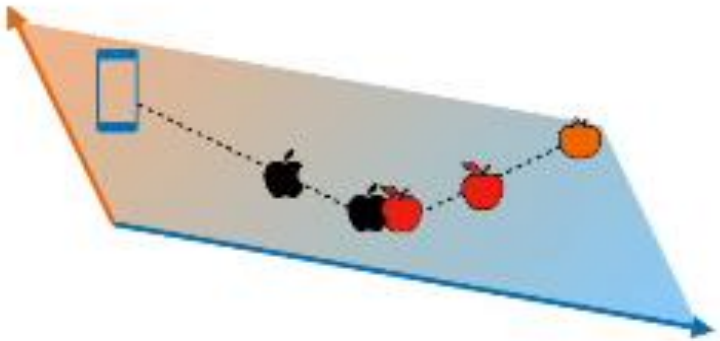
Linear transformations:



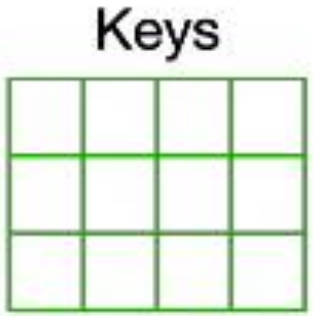
Okay

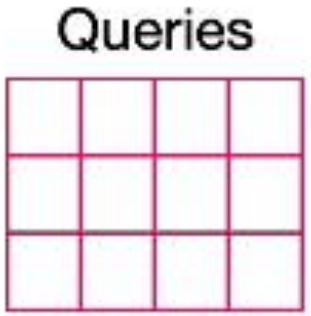


Bad



Good



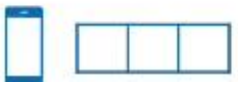


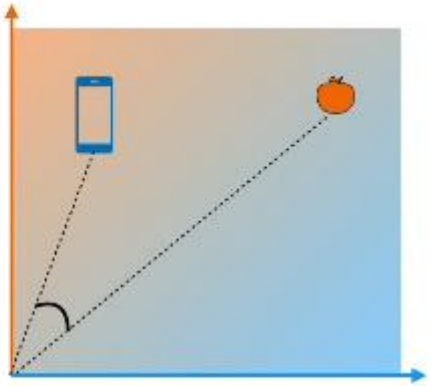
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

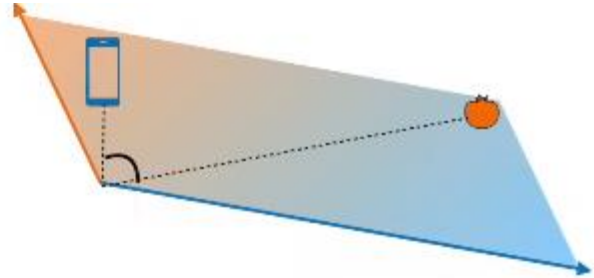
Keys, Query and Values **Transformer models**

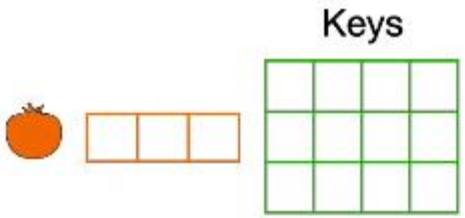
Similarity

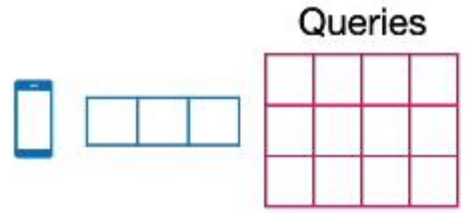


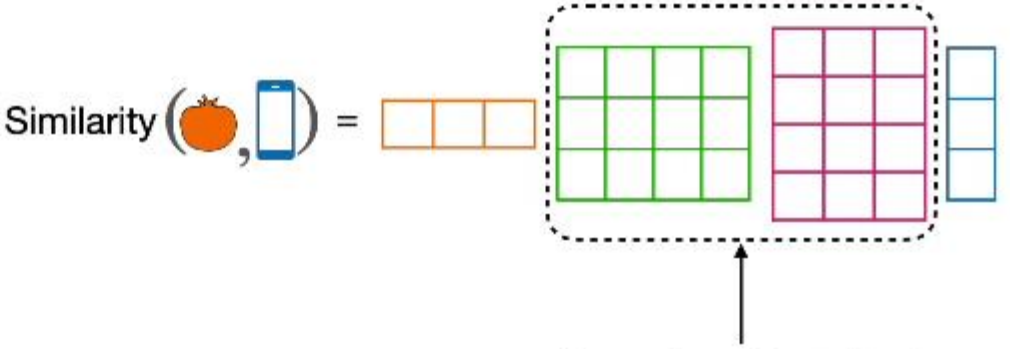


Keys and Queries Matrices







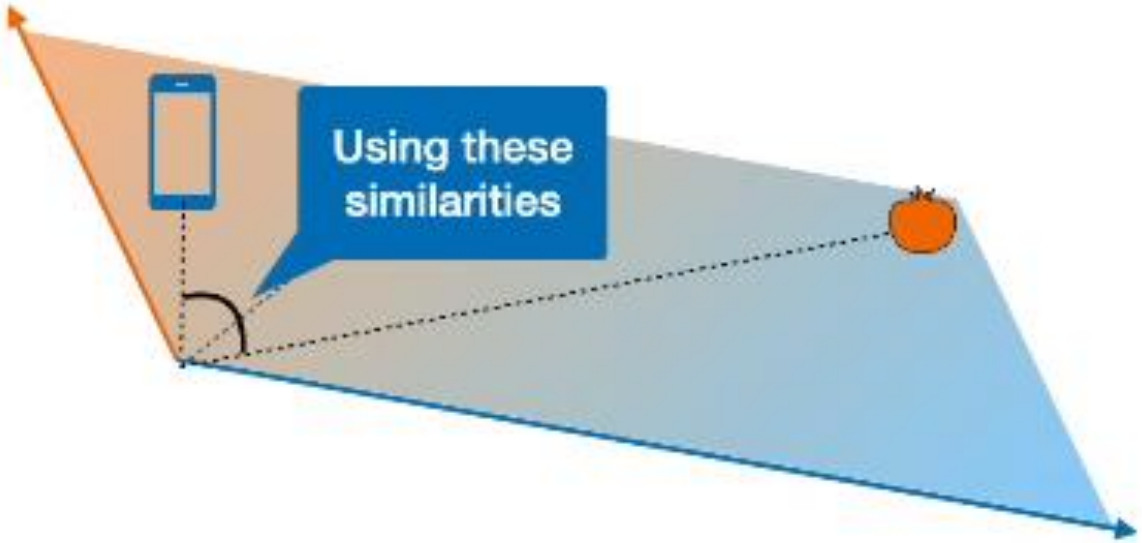


https://www.youtube.com/watch?v=UPtG_38Oc

Keys, Query and Values

Transformer models

Values matrix



Best embedding for finding similarities

Best embedding for finding the next word

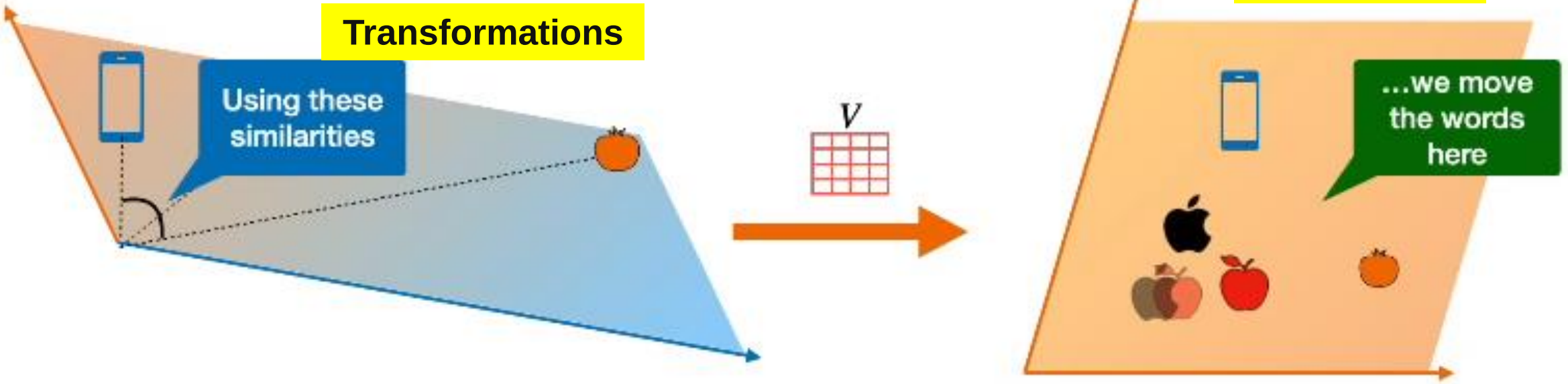
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Keys, Query and Values **Transformer models**

Values matrix

Transformations



Best embedding for finding similarities

Best embedding for finding the next word

Keys

Queries

Values

Embeddings here is responsible of capturing the color, size, technology, flavour, etc. (Features)

Embeddings here is responsible of capturing the next word in the sentence

https://www.youtube.com/watch?v=UPtG_38Oq8

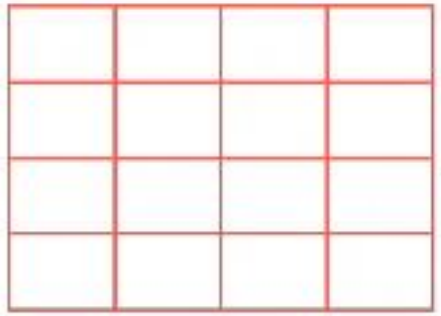
Keys, Query and Values Transformer models

Value matrix

an apple and an orange

	Orange	Apple	And	An
Orange	0.4	0.3	0.15	0.15
Apple	0.3	0.4	0.15	0.15
And	0.15	0.15	0.5	0.5
An	0.15	0.15	0.5	0.5

Value matrix



=

	Orange	Apple	And	An
Orange	v_{11}	v_{12}	v_{13}	v_{14}
Apple	v_{21}	v_{22}	v_{23}	v_{24}
And	v_{31}	v_{32}	v_{33}	v_{34}
An	v_{41}	v_{42}	v_{43}	v_{44}

apple $\longrightarrow$ $0.3 \cdot$ orange
$+0.4 \cdot$ apple
$+0.15 \cdot$ and
$+0.15 \cdot$ an

apple $\longrightarrow$ $v_{21} \cdot$ orange
$+v_{22} \cdot$ apple
$+v_{23} \cdot$ and
$+v_{24} \cdot$ an

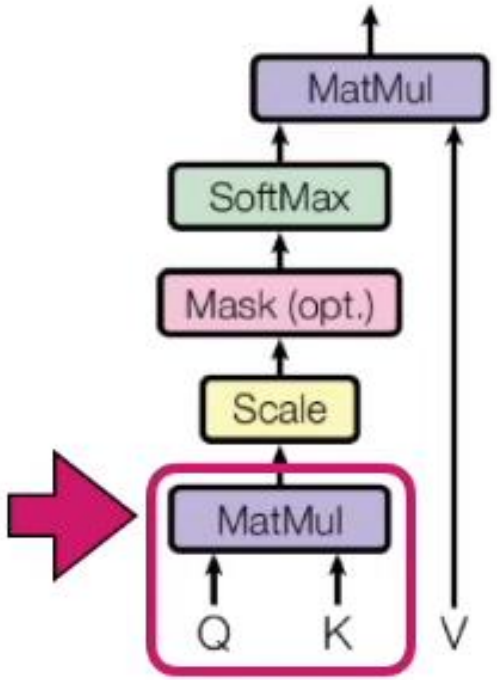
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

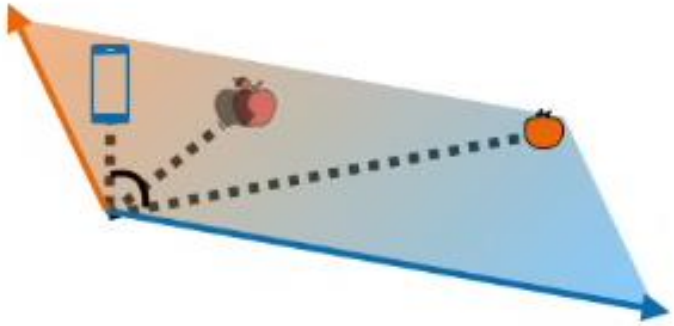
Attention Layer

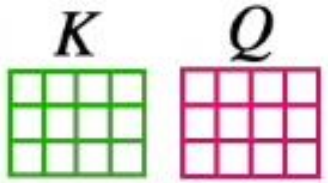
Transformer models

Scaled Dot-Product Attention



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





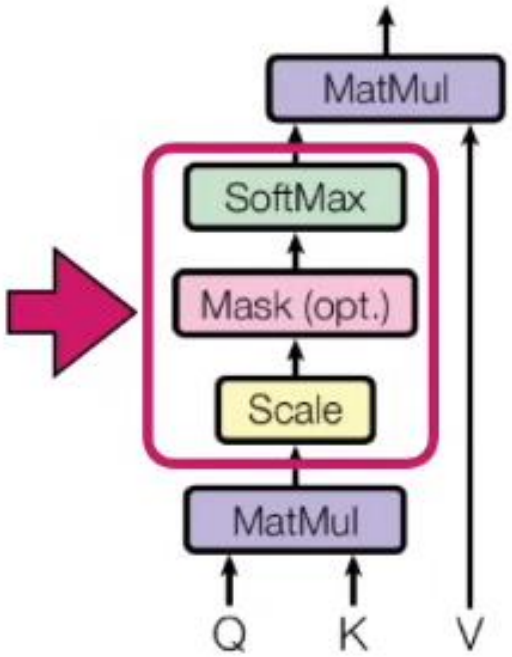
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

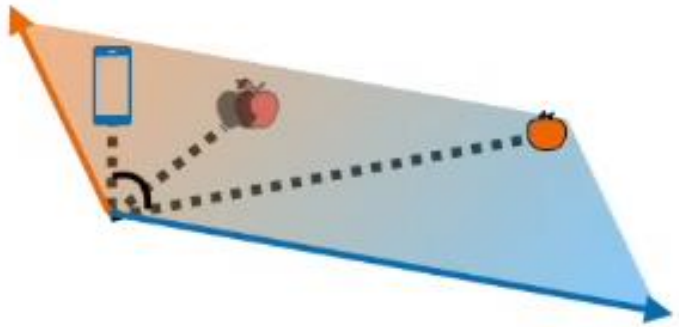
Attention Layer

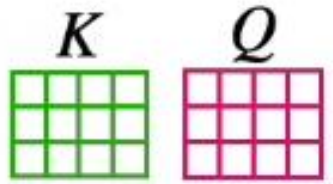
Transformer models

Scaled Dot-Product Attention



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





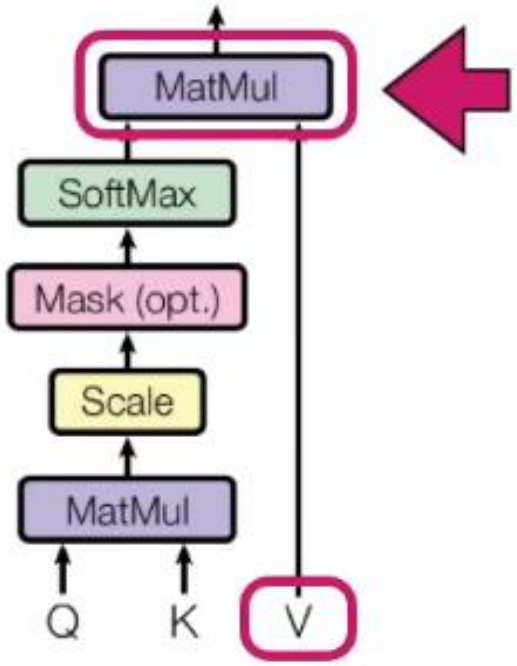
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

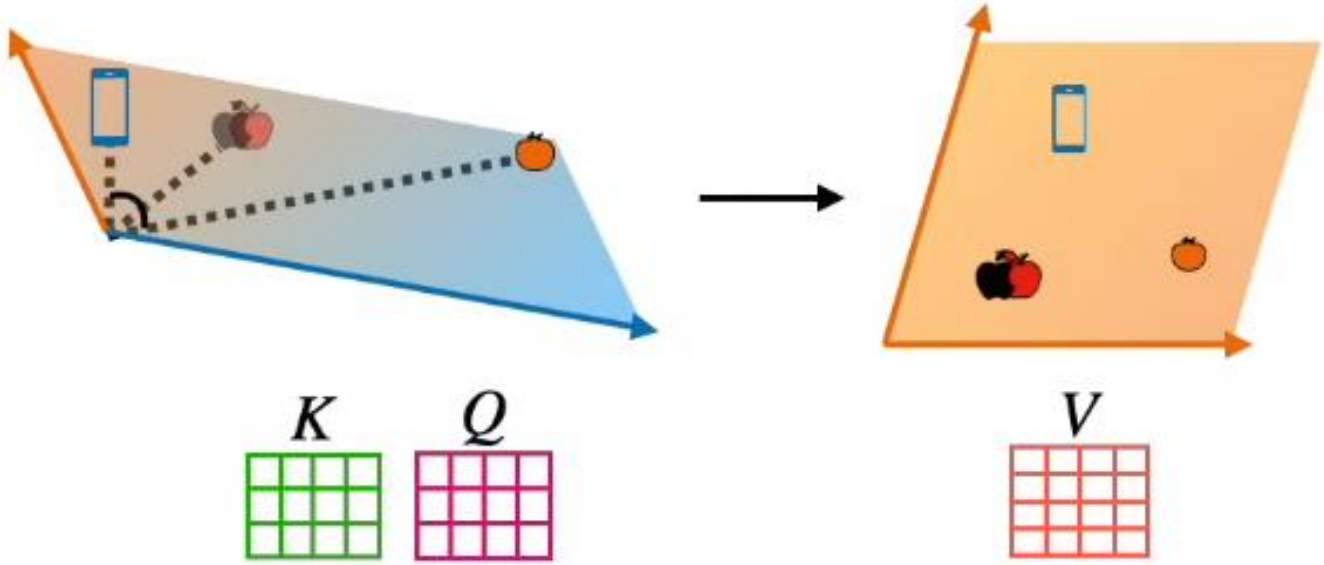
Attention Layer

Transformer models

Scaled Dot-Product Attention



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

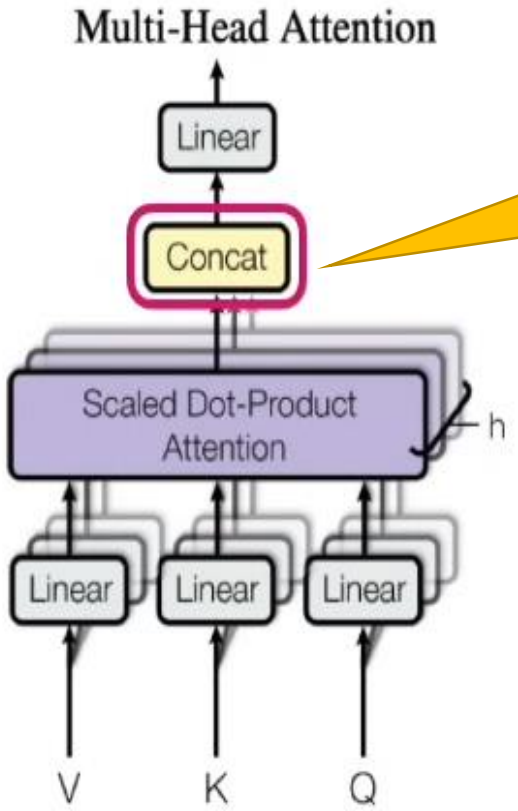


https://www.youtube.com/watch?v=UPtG_38Oq8o

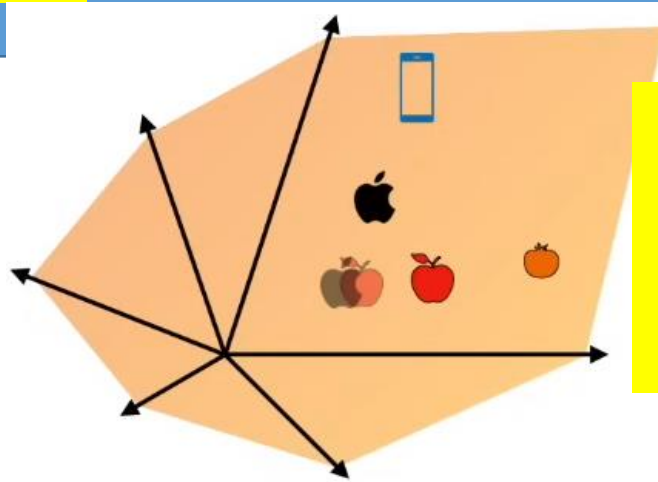
Language Models

Multi-Head Attention Layer

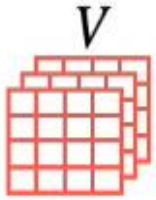
Transformer models

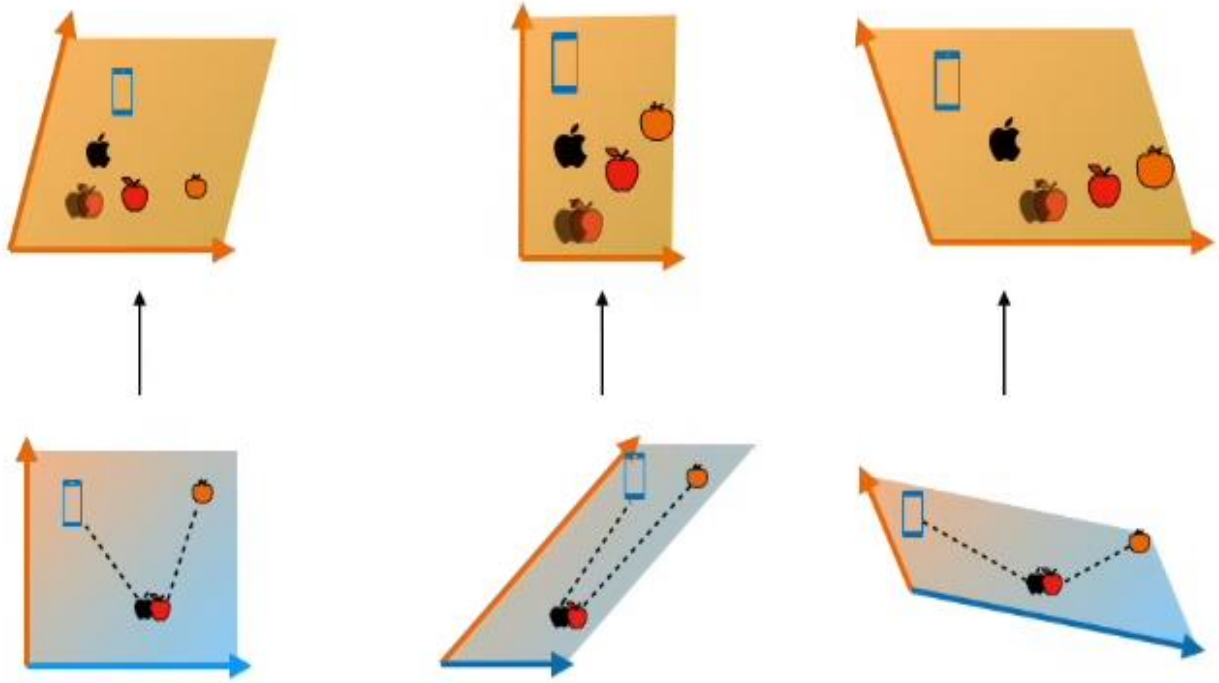


Concatenation of all embeddings since we don't know which one is better!



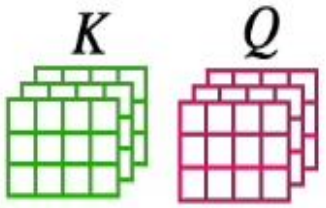
Getting higher dimensionality because of concatenation layer





$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_h)W^O$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

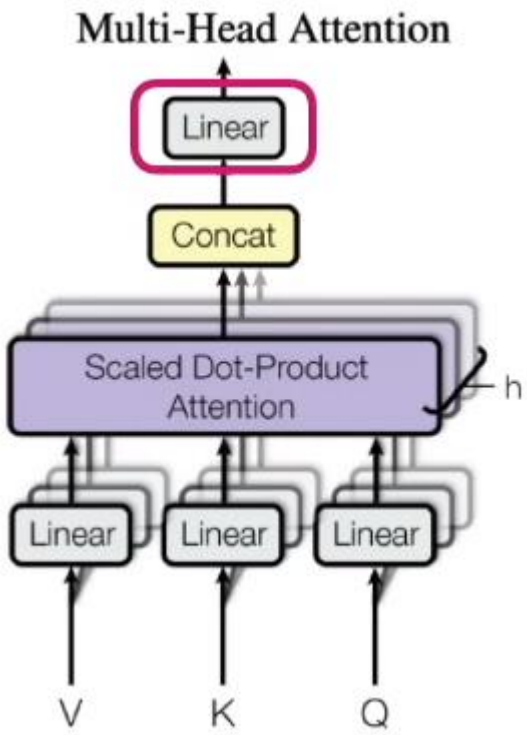


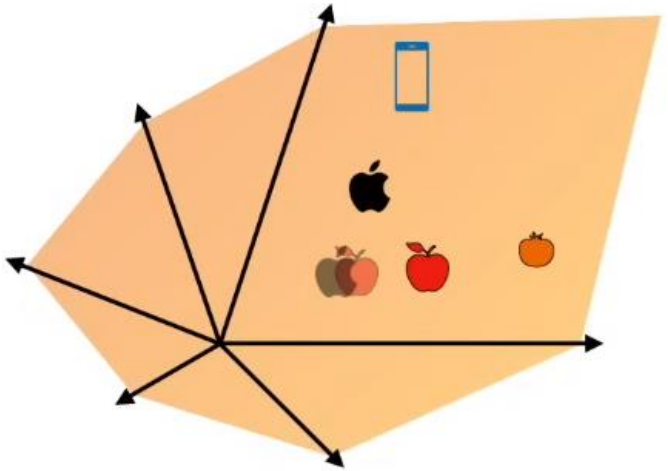
https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

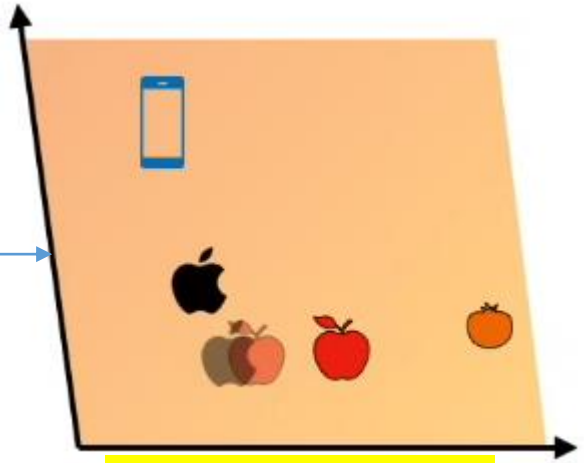
Multi-Head Attention Layer

Transformer models





Linear transformation (Rectangular matrix multiplication)



Optimal representation

https://www.youtube.com/watch?v=UPtG_38Oq8o

Language Models

Transformer models

How are you

Write a story.

Write a story. Once

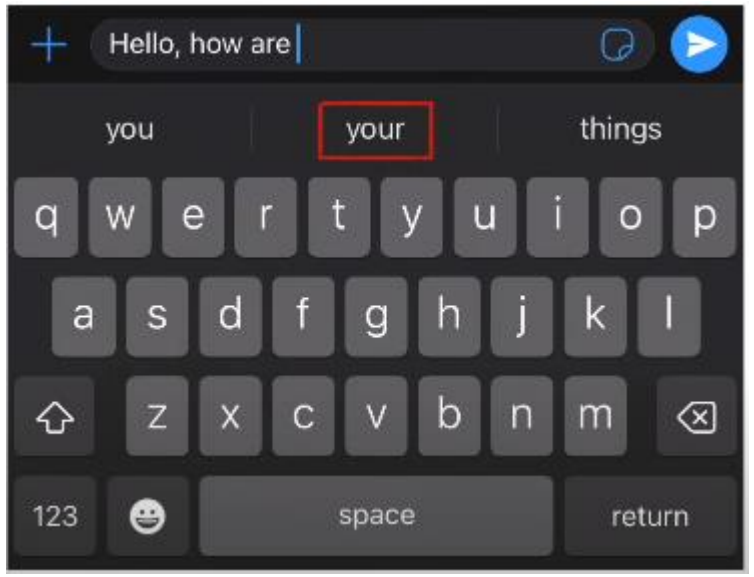
Write a story. Once upon

doing

Once

upon

a



https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

ngrams

Transformer models

Ngram: method of Looking at N words that appears before!

1gram example

Hello, How are

Data

… are **you** …

… are **sad** …

… are**?**…

… are **happy** …

… are **ready** …

… are **happy** …

… are **free** …

3grams example

Hello, How are

Data

Hello, how are **you**?

Hello, how are **things** going?

Hello, how are **things** today?

Hello, how are **the** kids?

Hello, how are **the** others?

Hello, how are **they** doing?

Hello, how are **things** happening?

That day where I was singing and two clouds appeared ……..

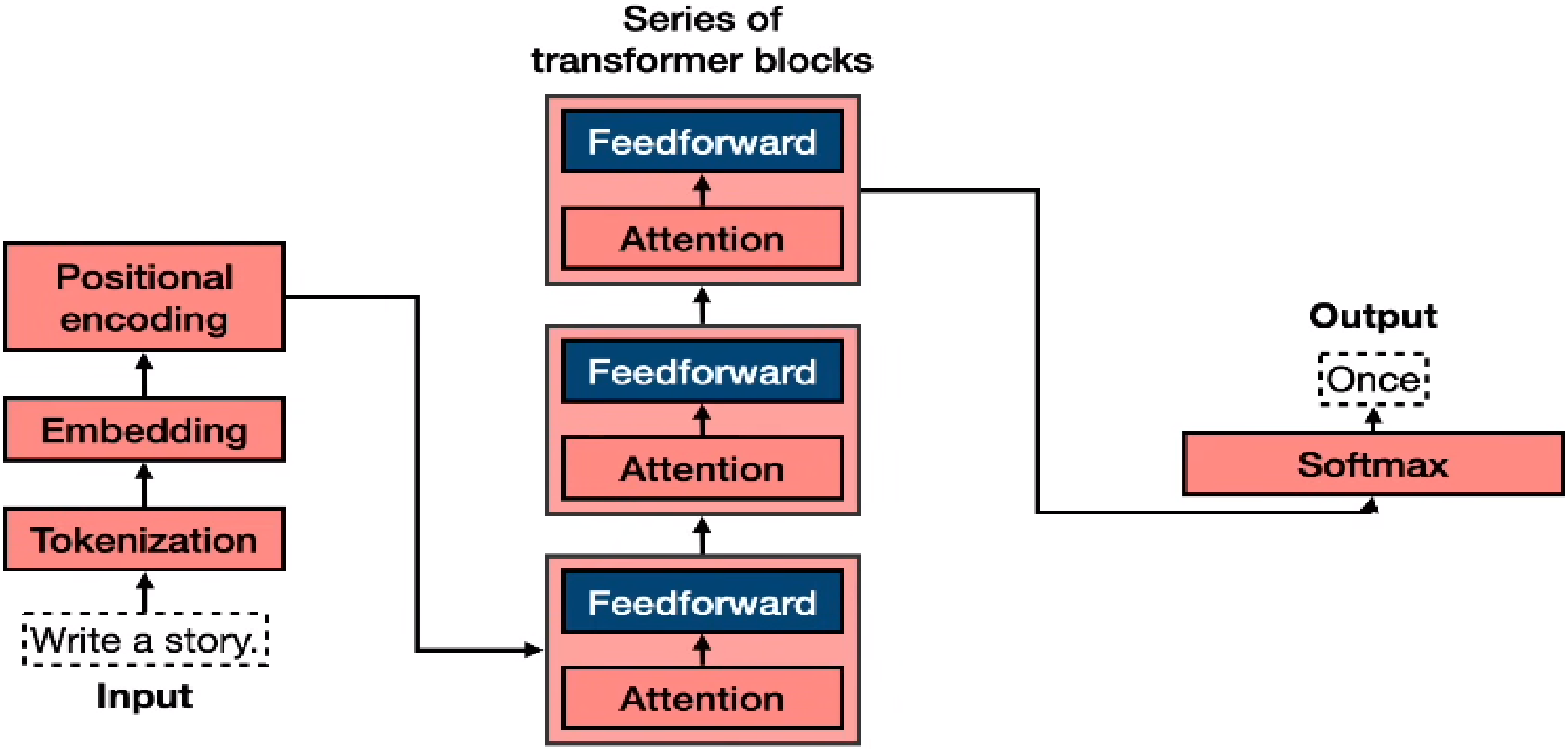
Data

???

https://www.youtube.com/watch?v=Ppf4ApbumcA

Language Models

Transformer models



Language Models

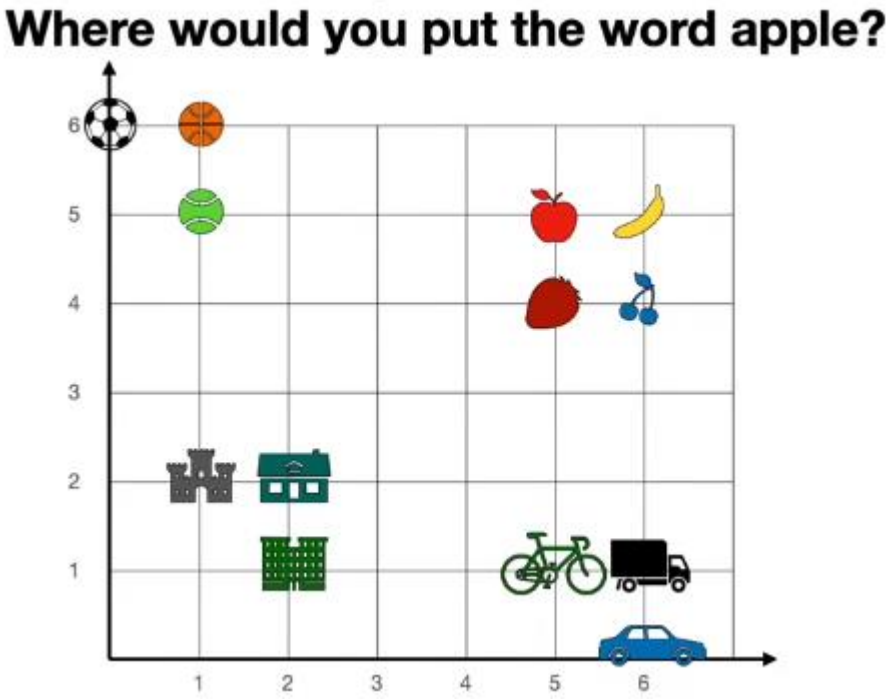
Transformer models

Step 1: Tokenization
Split sentence into segments (words or tokens)

Write a story. → Write | A | story | .

doesn't → does | n't

Step 2: Embeddings



Word	Numbers	
Apple	5	5
Banana	6	5
Strawberry	5	4
Cherry	6	4
Soccer	0	6
Basketball	1	6
Tennis	1	5
Castle	1	2
House	2	2
Building	2	1
Bicycle	5	1
Truck	6	1
Car	6	0

https://www.youtube.com/watch?v=Ppf4ApbumcA

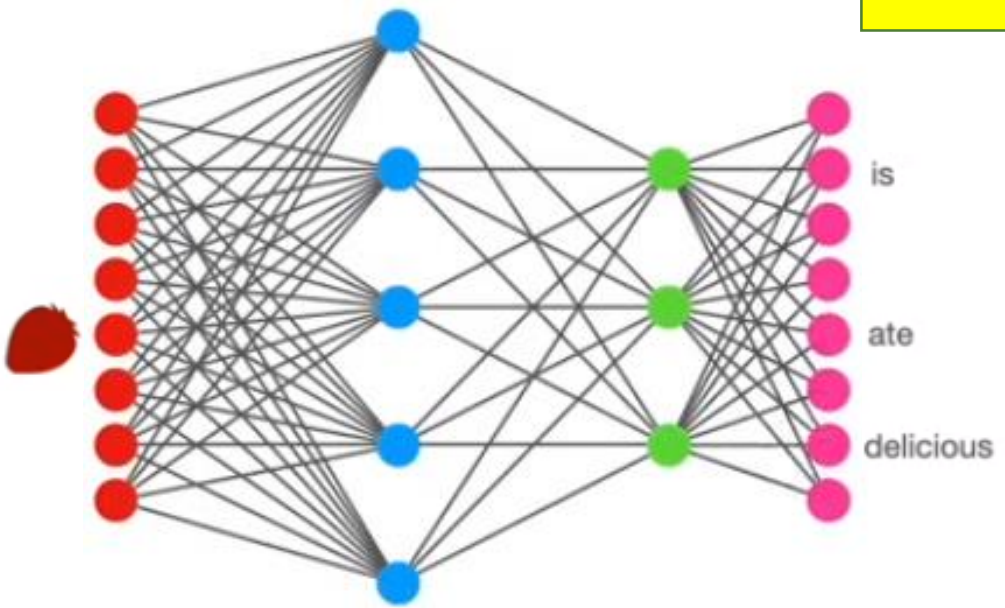
Language Models

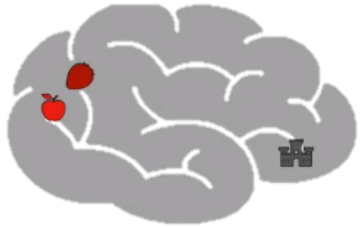
Step 2: Embeddings

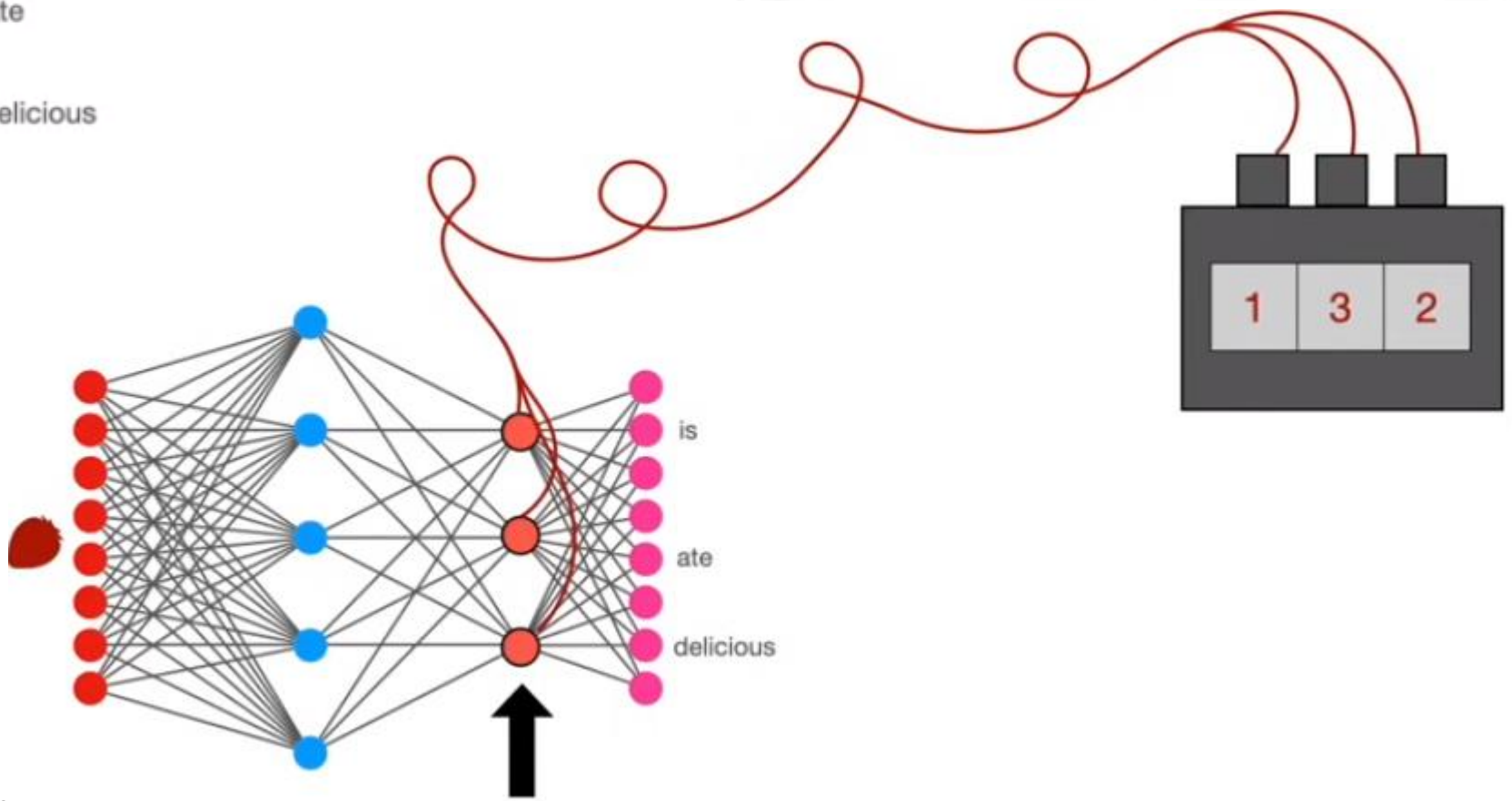
Word2Vec embedding method

Transformer models

Word	Numbers		
Strawberry	1	3	2
Apple	1.1	2.9	2.2
Castle	7	-5.4	0.4







https://www.youtube.com/watch?v=Ppf4Apbumca

Language Models

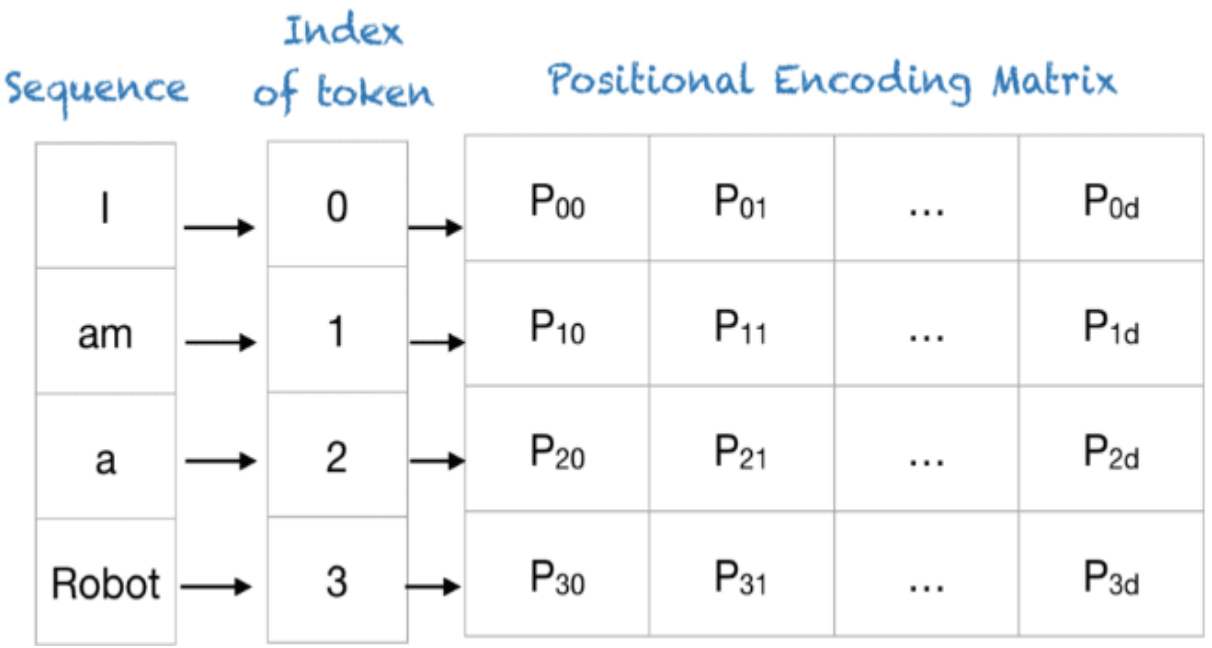
Step 3: Positional Encoding

Transformer models

What is positional encoding (first part of encoder)

Index value, can't be used to represent an item's position in transformer models (For long sequences, the indices can grow large in magnitude).
Normalize the index value to lie between 0 and 1 can create problems for variable length sequences as they would be normalized differently.

Transformer used smart positional encoding.
Describes the location or ***position*** of an entity in a sequence so that each position is assigned a ***unique representation***.
Each position/index is mapped to a vector.
output of the positional encoding layer is a matrix, where each row of the matrix represents an encoded object of the sequence summed with its positional information.



https://www.youtube.com/watch?v=Ppf4ApbumcA

Positional Encoding Concept

Taking into account the ***order*** or words in the context.

Let's assume that we have an input sequence of length L

We need to know the position of the K^{th} object (word).

The positional encoding is given by sine and cosine functions of varying frequencies:

$$P(k, 2i) = \sin\left(\frac{k}{n^{2i/d}}\right)$$

$$P(k, 2i+1) = \cos\left(\frac{k}{n^{2i/d}}\right)$$

P(k,j): ***Position function*** for mapping a position ***K*** in the input sequence to index ***(k,j)*** of the positional matrix

d: Dimension of the output embedding space

n: User-defined scalar, set to 10,000 by the authors of Attention Is All You Need.

$0 <= i < d/2$

Positional Encoding *Example "I am a robot"* with n=100 and d=4

$$P(k, 2i) = \sin\left(\frac{k}{n^{2i/d}}\right)$$

$$P(k, 2i+1) = \cos\left(\frac{k}{n^{2i/d}}\right)$$

Positional Encoding Matrix with d=4, n=100

Sequence	Index of token, k	i=0	i=0	i=1	i=1
I	0	$P_{00}=\sin(0)$ = 0	$P_{01}=\cos(0)$ = 1	$P_{02}=\sin(0)$ = 0	$P_{03}=\cos(0)$ = 1
am	1	$P_{10}=\sin(1/1)$ = 0.84	$P_{11}=\cos(1/1)$ = 0.54	$P_{12}=\sin(1/10)$ = 0.10	$P_{13}=\cos(1/10)$ = 1.0
a	2	$P_{20}=\sin(2/1)$ = 0.91	$P_{21}=\cos(2/1)$ = -0.42	$P_{22}=\sin(2/10)$ = 0.20	$P_{23}=\cos(2/10)$ = 0.98
Robot	3	$P_{30}=\sin(3/1)$ = 0.14	$P_{31}=\cos(3/1)$ = -0.99	$P_{32}=\sin(3/10)$ = 0.30	$P_{33}=\cos(3/10)$ = 0.96

Positional Encoding *Python implementation*

```python
import numpy as np
import matplotlib.pyplot as plt

def getPositionEncoding(seq_len, d, n=10000):
    P = np.zeros((seq_len, d))
    for k in range(seq_len):
        for i in np.arange(int(d/2)):
            denominator = np.power(n, 2*i/d)
            P[k, 2*i] = np.sin(k/denominator)
            P[k, 2*i+1] = np.cos(k/denominator)
    return P

P = getPositionEncoding(seq_len=4, d=4, n=100)
print(P)
```

$$P(k, 2i) = \sin\left(\frac{k}{n^{2i/d}}\right)$$

$$P(k, 2i+1) = \cos\left(\frac{k}{n^{2i/d}}\right)$$

```
[[ 0.          1.          0.          1.        ]
 [ 0.84147098  0.54030231  0.09983342  0.99500417]
 [ 0.90929743 -0.41614684  0.19866933  0.98006658]
 [ 0.14112001 -0.9899925   0.29552021  0.95533649]]
```

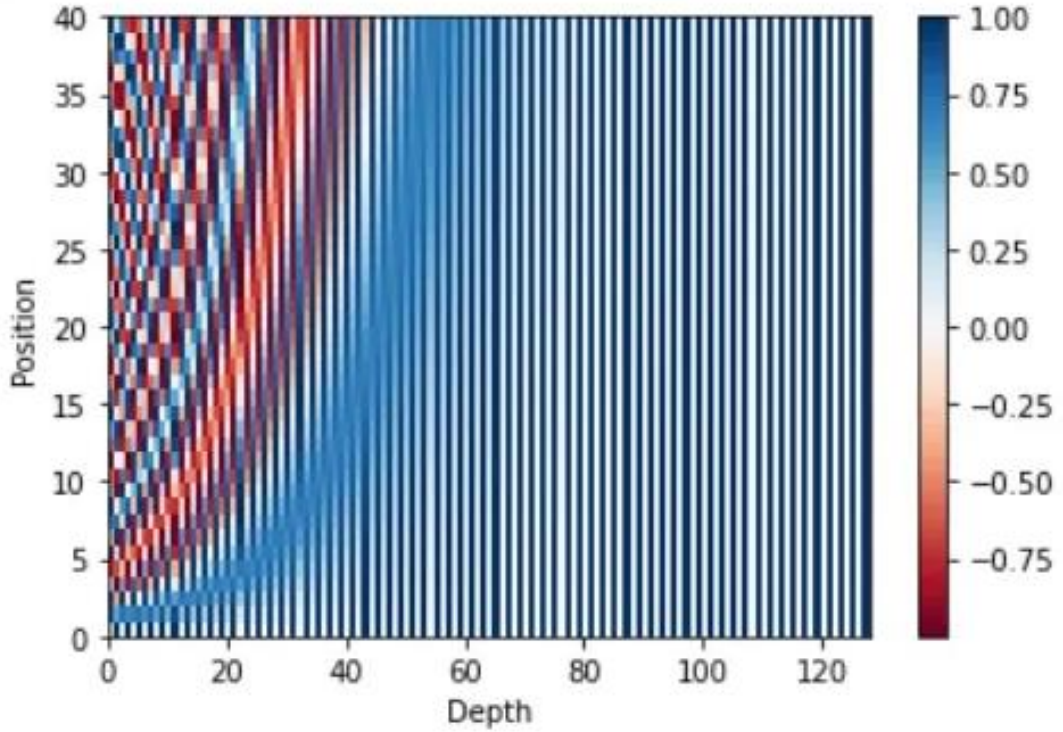
A Gentle Introduction to Positional Encoding in Transformer Models, Part 1 - MachineLearningMastery.c

Positional Encoding *(Why sin and cosine)*

1.The sine and cosine functions have values in [-1, 1], which keeps the values of the positional encoding matrix in a normalized range.

2.As the sinusoid for each position is different, you have a unique way of encoding each position.

3.You have a way of measuring or quantifying the similarity between different positions, hence enabling you to encode the relative positions of words.



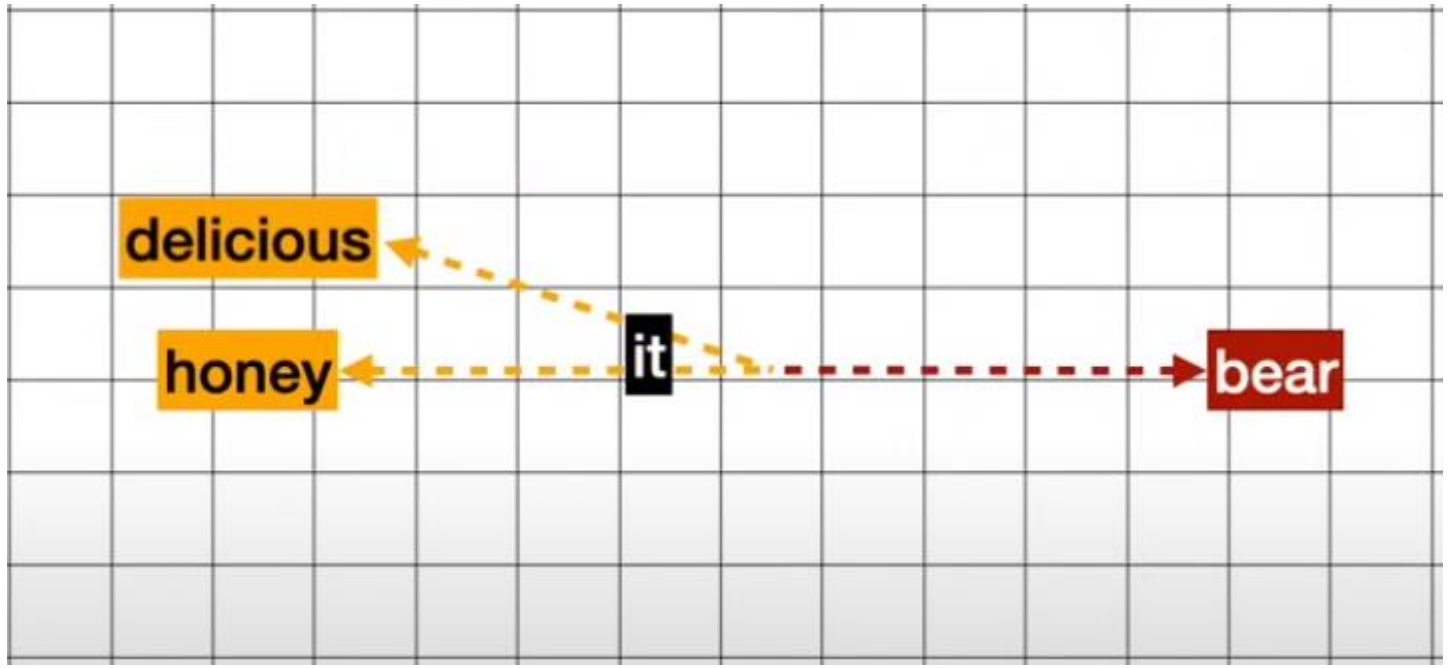
Understanding Positional Encoding in Transformers | by Alvaro Henriquez | Analytics Vidhya | Medium

Capturing the context by make attention to specific parts

Delicious

The bear ate the honey because it was ……..

Hungry

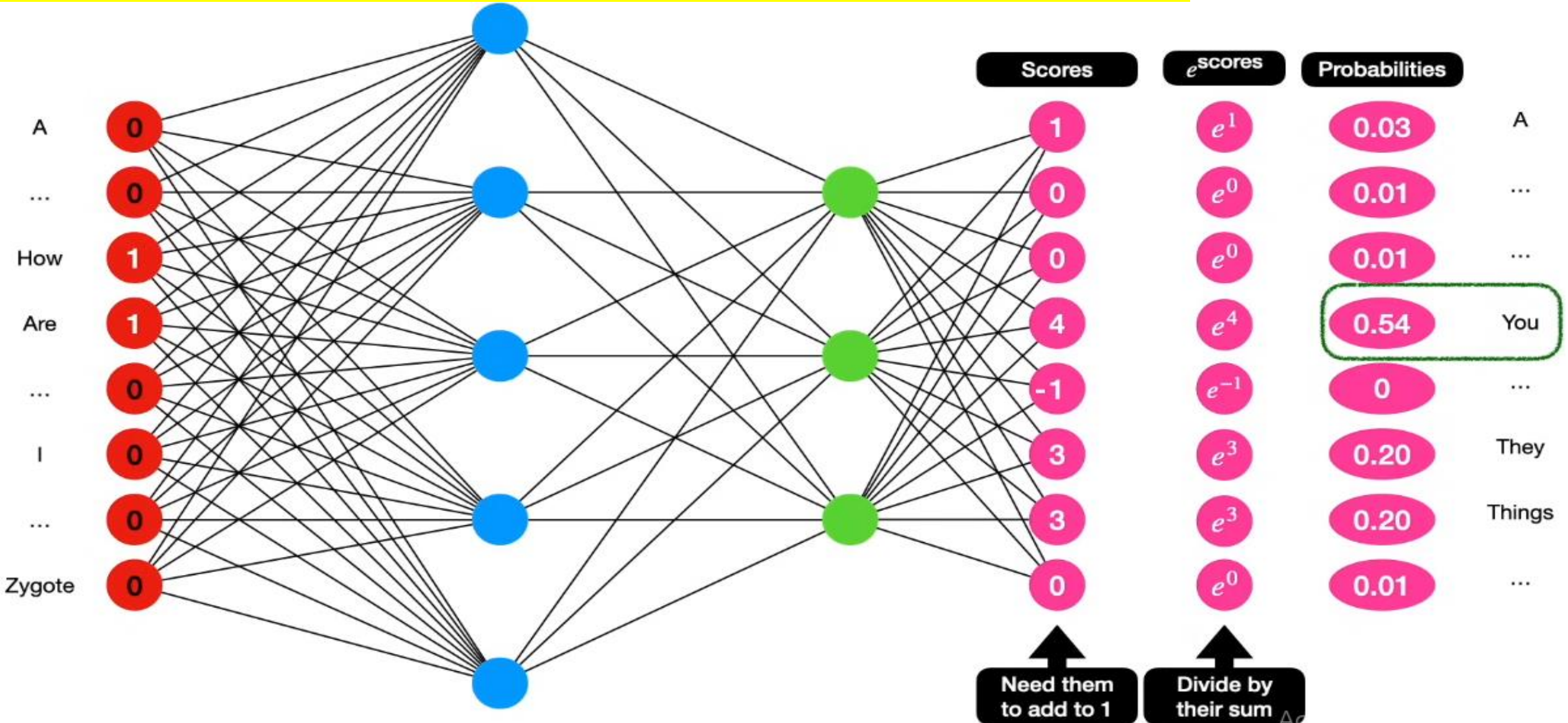


Language Models

Step 5: Softmax

Transformer models

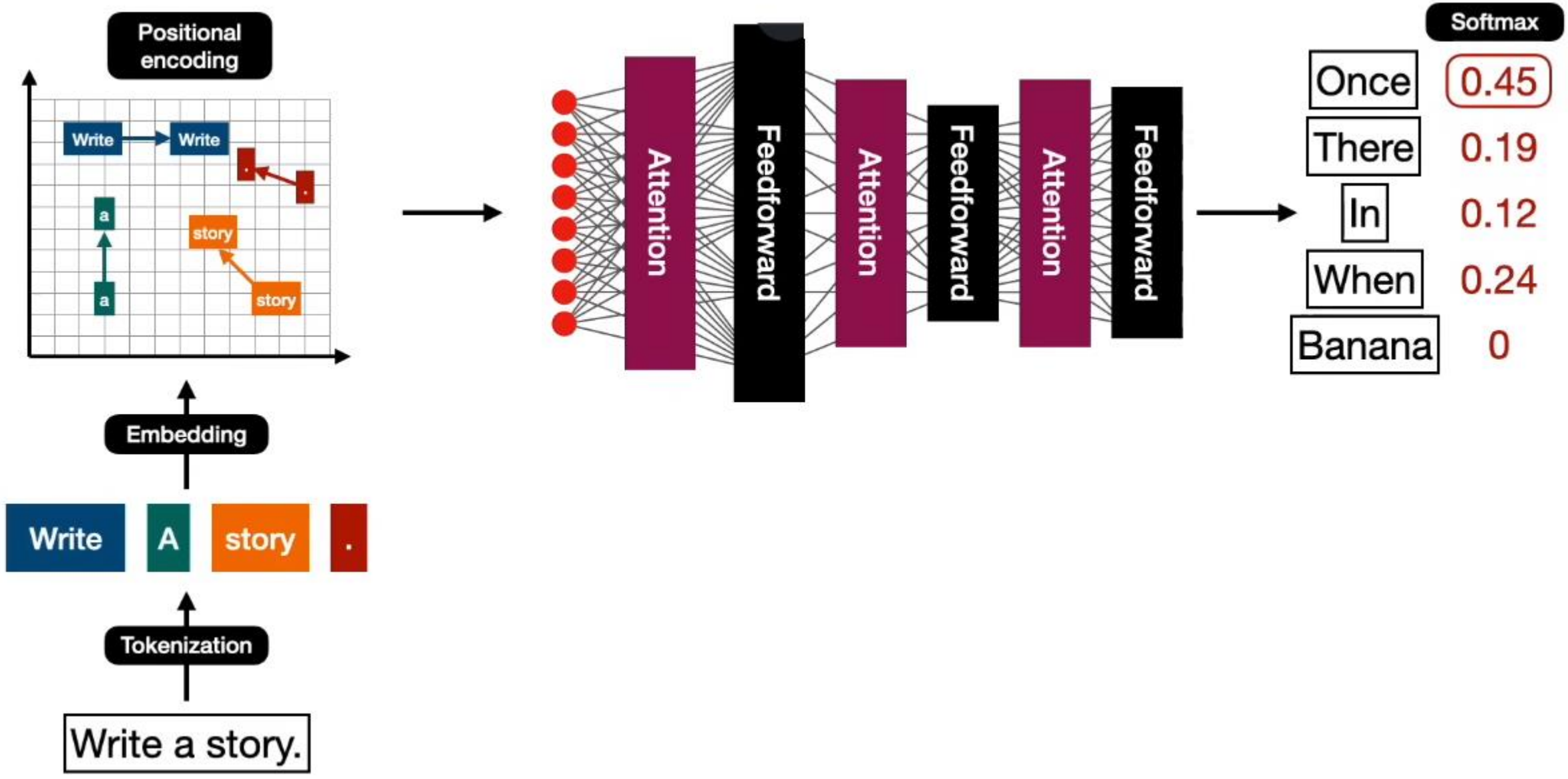
Classification based on Probability



Language Models

Putting all together

Transformer models



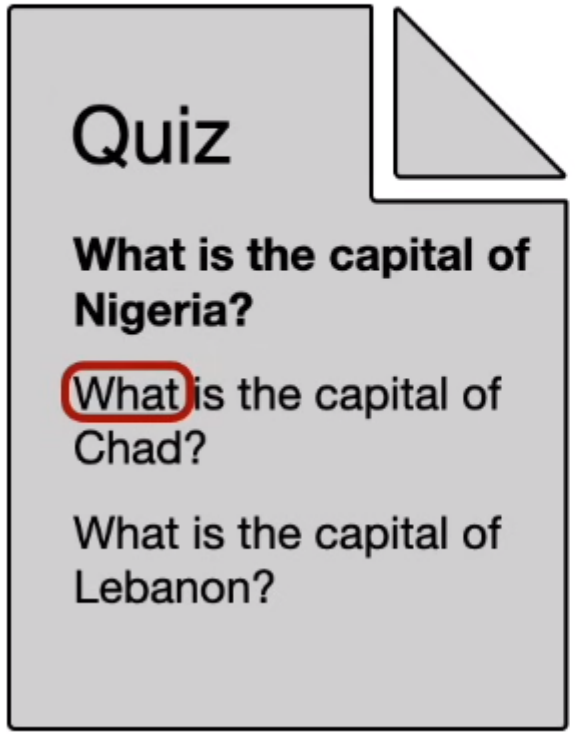
Language Models

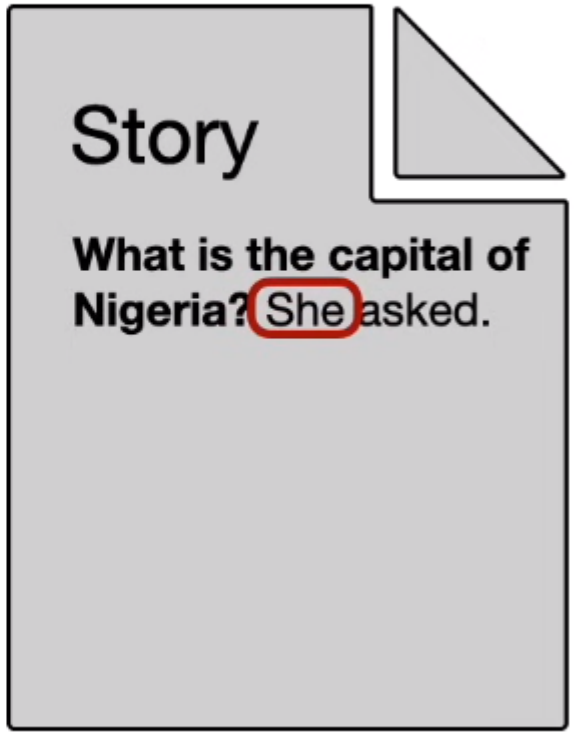
Transformer models

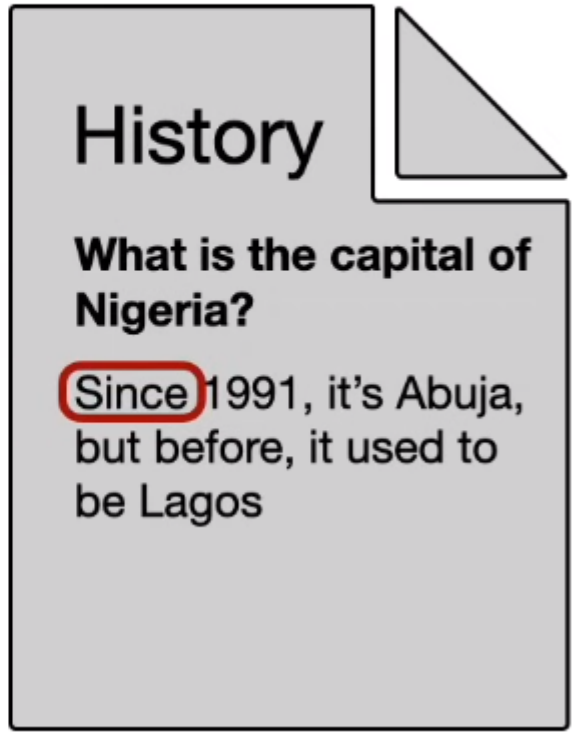
The transformer models needs to be trained on large datasets to learn how to predict the next word in the sentence.

Now what if we want the transformer model to answer a question?

What is the capital of Nigeria? **Abuja**







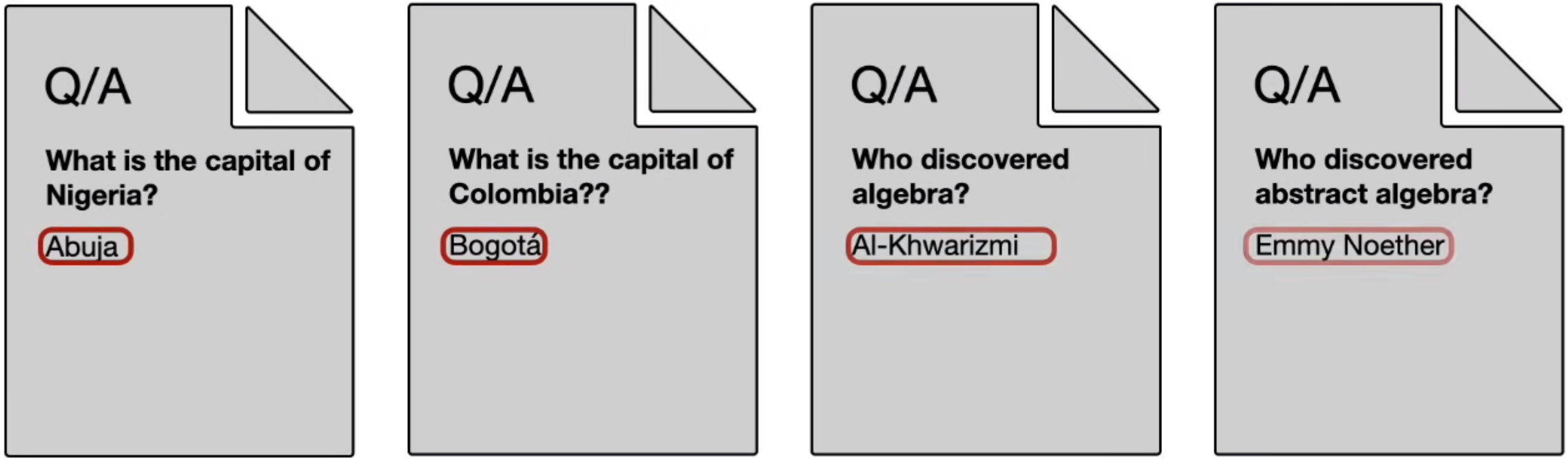
Language Models

Transformer models

How to enforce transformer model to answer questions?

Post training (Fine tuning)
Training the transformer model on a dataset of questions and answers

What is the capital of Nigeria? **Abuja**



Language Models

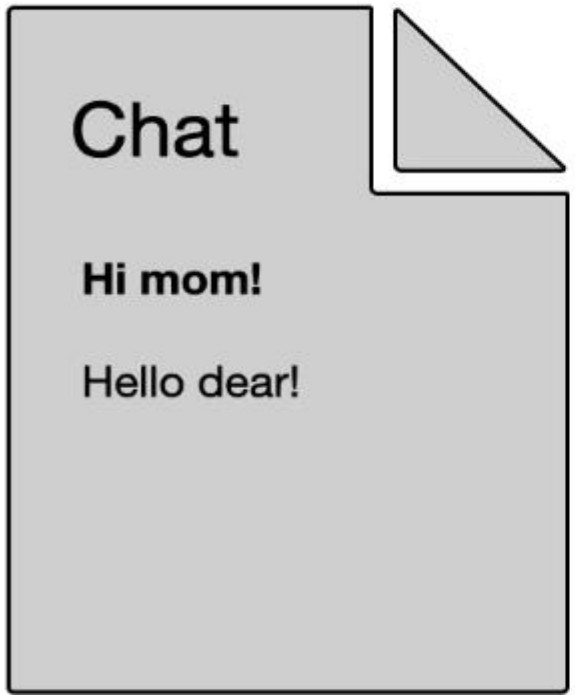
Transformer models

How to enforce transformer model to chat with us?

Post training (Fine tuning)

Training the transformer model on a dataset of *commands and actions*

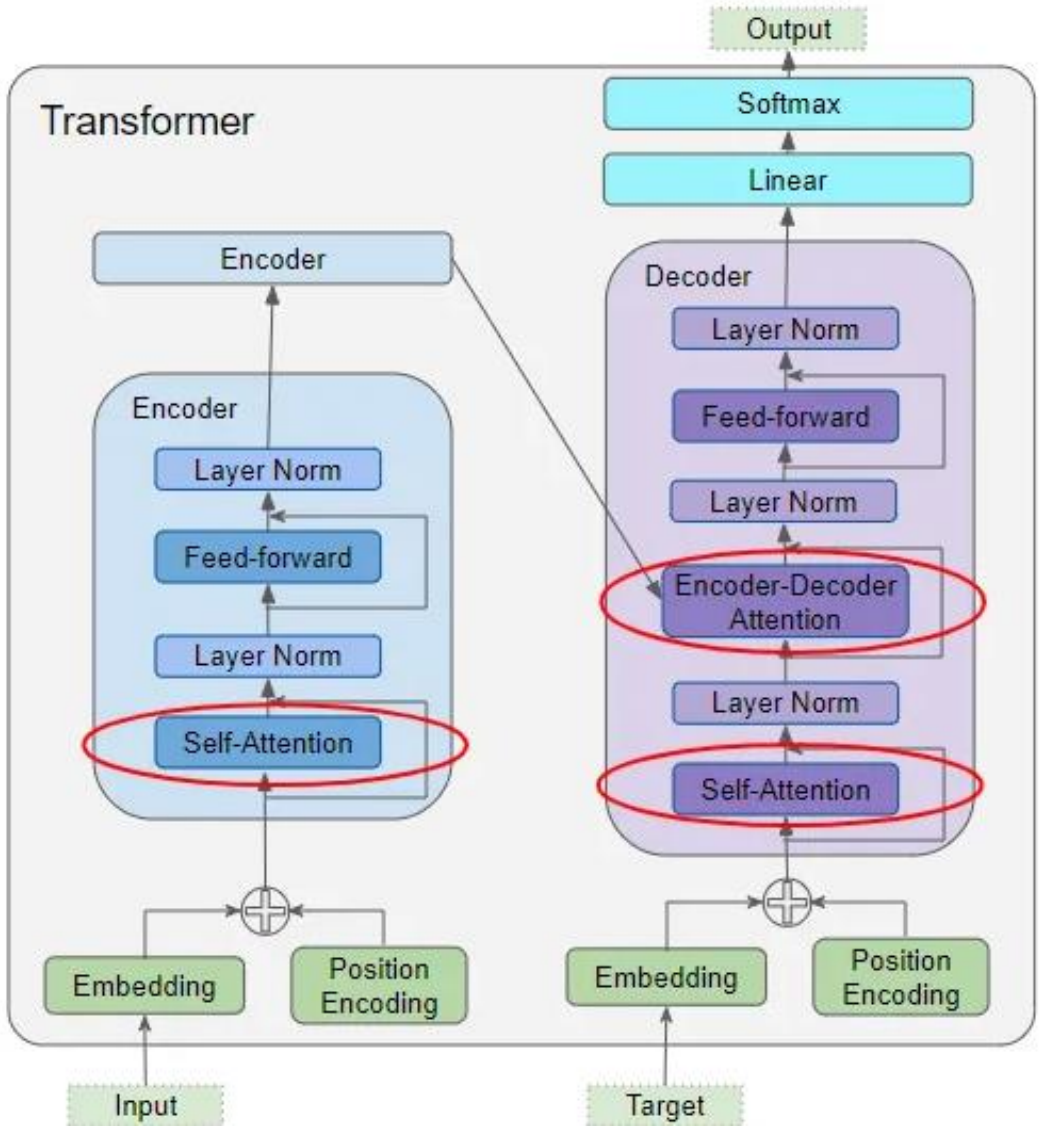
Hello, how are you?
Good, and you?
...



Language Models

Transformer models

When to use both parts of transformer model (Encoder & Decoder)



In transformation applications:
Translation from French to English

In this case, we need an extra Encoder-Decoder attention layer
We also need to connect the output of the last encoder part with this Encoder-Decoder attention layer of the first decoder part.