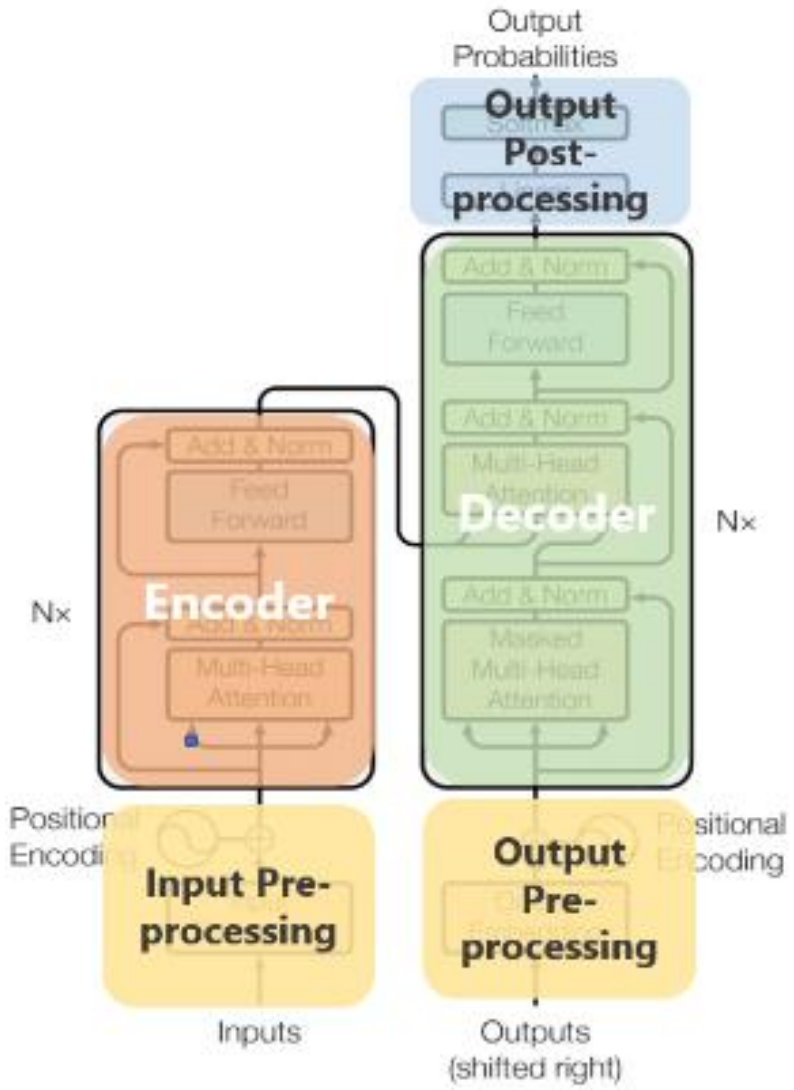
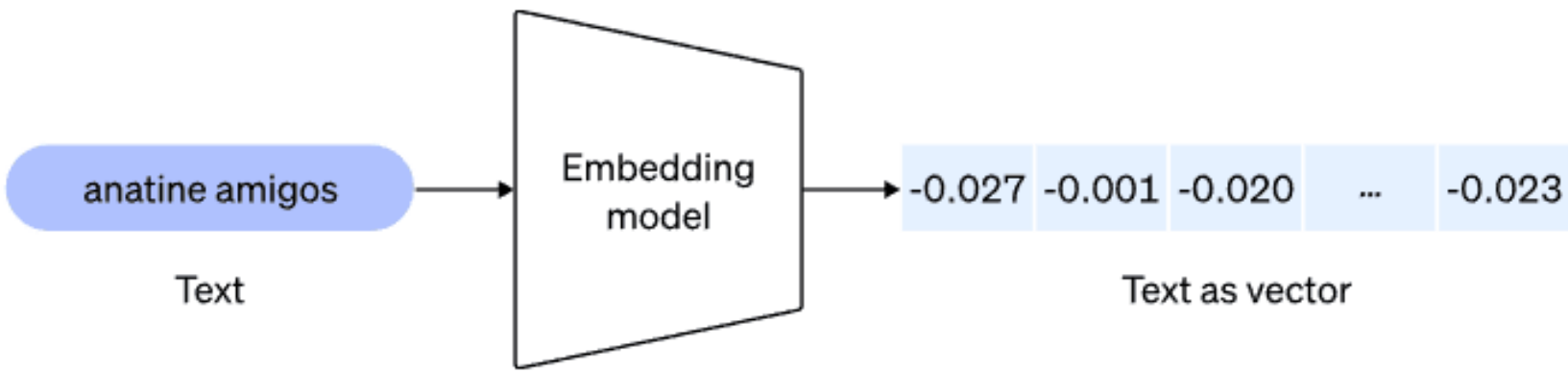
INTRO TO TRANSFORMERS



Embeddings

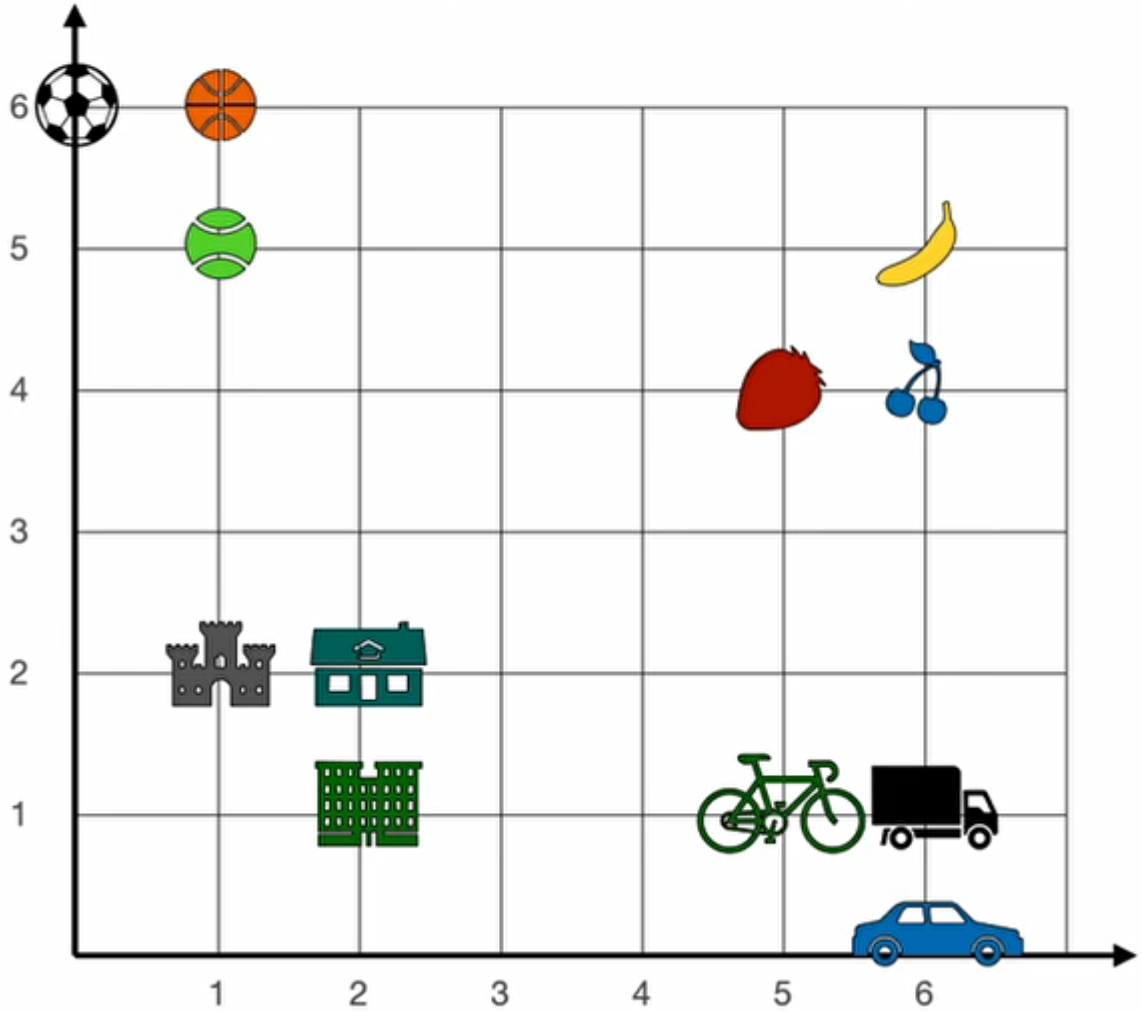
Embeddings


PALABRAS
LO
ESTOY
YA
MIS
TIENEN
es
que
PUE
Tagxedo
si
UNA
huestras
POCO
MUY
NUES


anatine amigos
Text
Embedding model
-0.027
-0.001
-0.020
...
-0.023
Text as vector

Embeddings Quiz 1

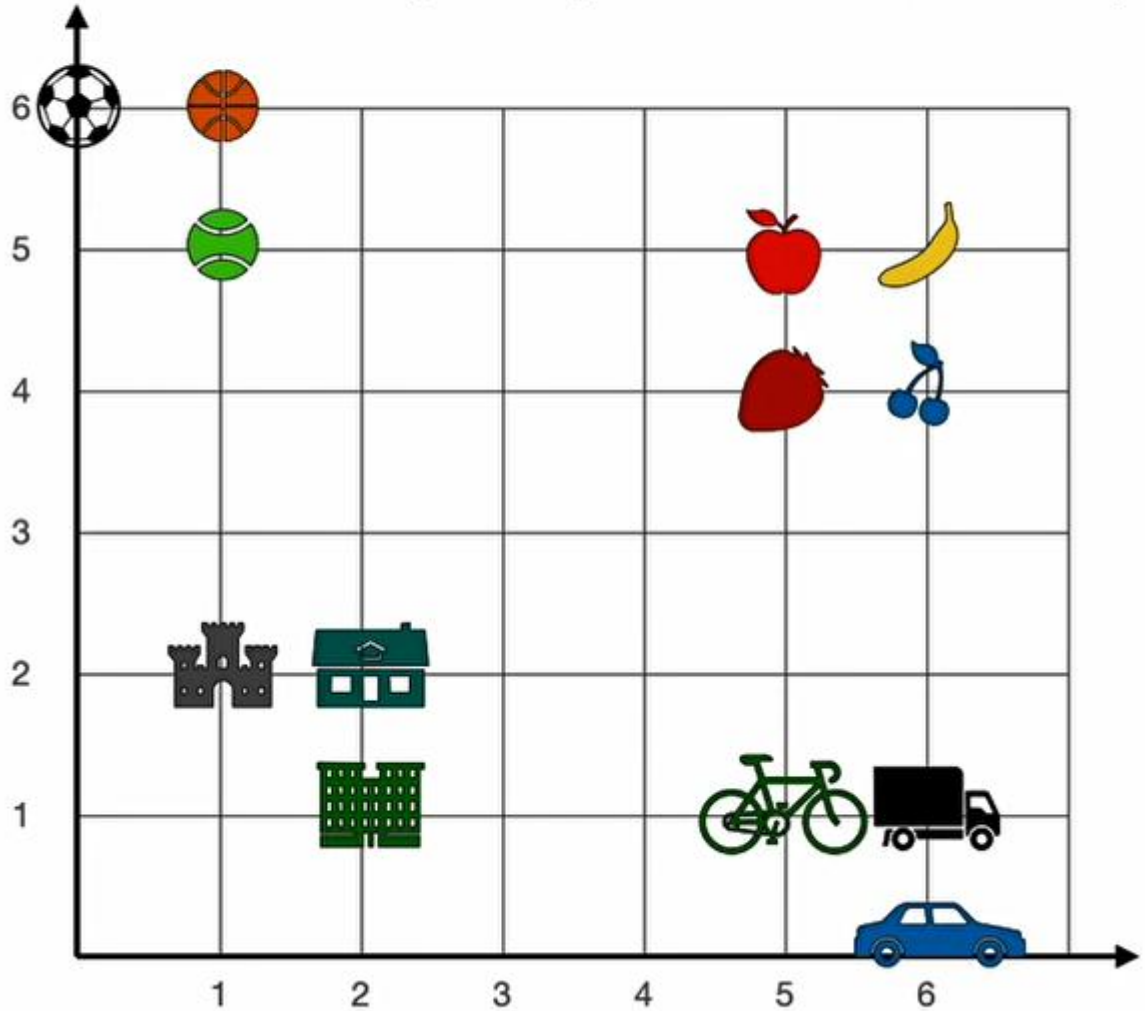
Where would you put the word apple?



Word	Numbers	
Apple	?	?
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

Embeddings Quiz 1

Where would you put the word apple?

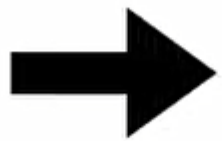


Word	Numbers	
Apple	5	5
Banana	6	5
Strawberry	5	4
Cherry	6	4
Soccer	0	6
Basketball	1	6
Tennis	1	5
Castle	1	2
House	2	2
Building	2	1
Bicycle	5	1
Truck	6	1
Car	6	0

Word embeddings

Many more columns

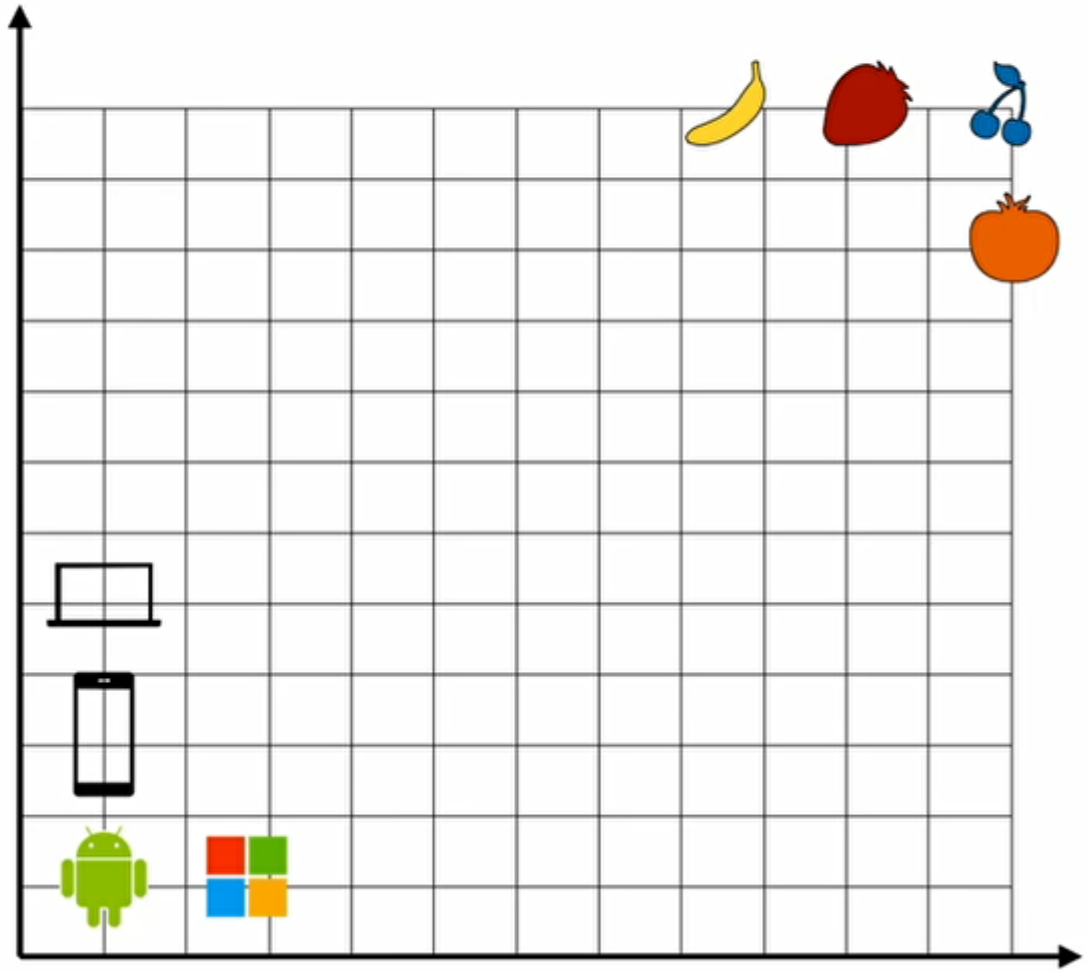
Word	Numbers	
Apple	5	5
Soccer	0	6
House	2	2
Car	6	0



Word	Numbers			
A	-0.82	-0.32	…	0.23
Aardvark	0.419	1.28	…	-0.06
…			…	
Zygote	-0.74	-1.02	…	1.35

4096

Embeddings Quiz 2



Top right or bottom left?

Cherry

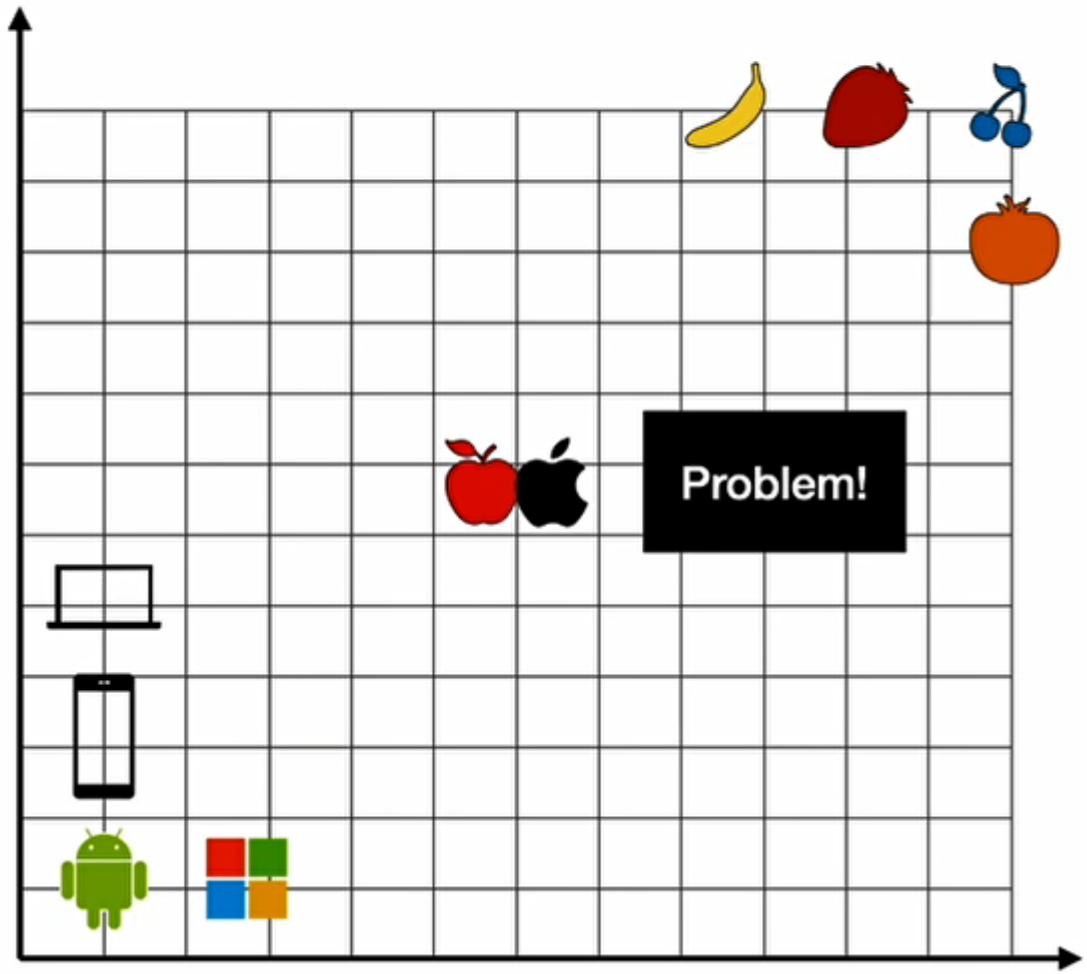
Android

Laptop

Banana

Apple?

Embeddings Quiz 2



Top right or bottom left?

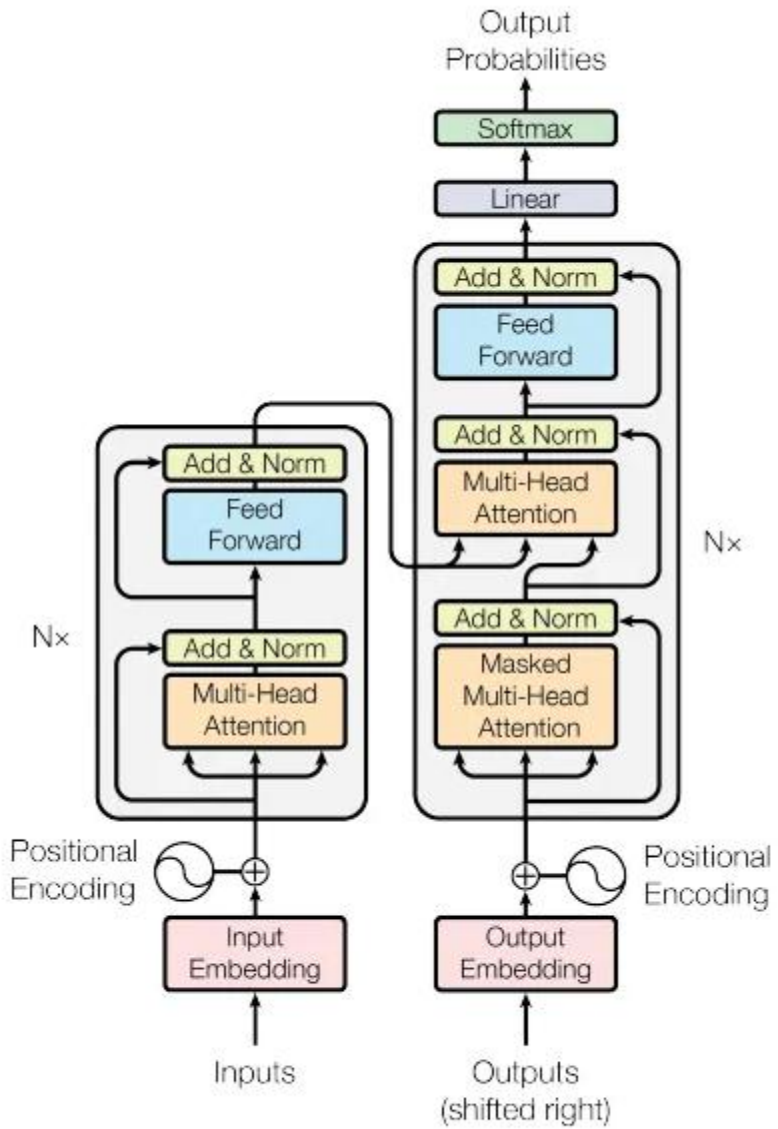
Cherry

Android

Laptop

Banana

Apple?



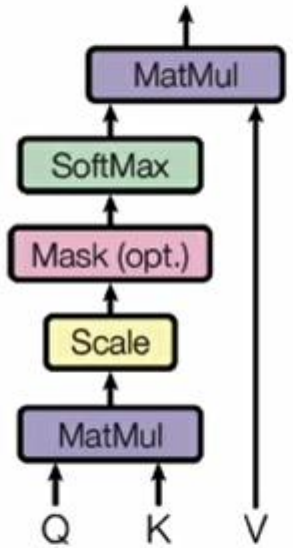
From "Attention is all you need" paper by Vaswani, et al., 2017 [1]

WE CAN SOLVE THIS TYPE OF PROBLEM WITH ATTENTION, SELF ATTENTION, IT USE THE CONTENT OF THE SENTENCE IN ORDER TO HELP THE EMBEDDINGS TO RISOLVE THIS TYPE OF AMBIGUOUS

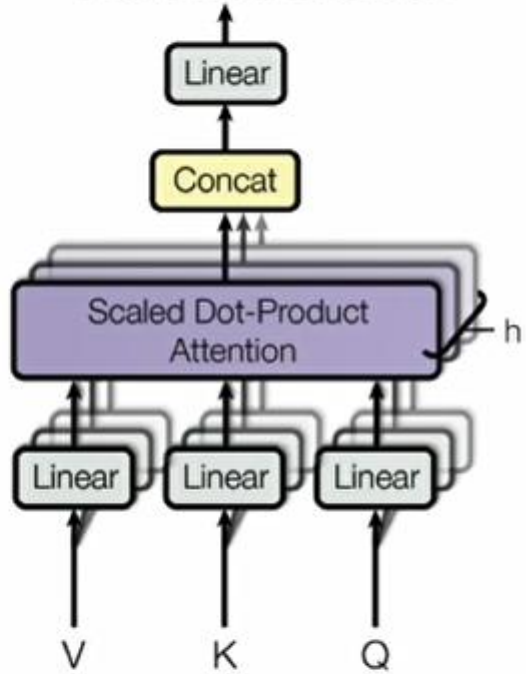
Self- Attention

Attention

Scaled Dot-Product Attention



Multi-Head Attention

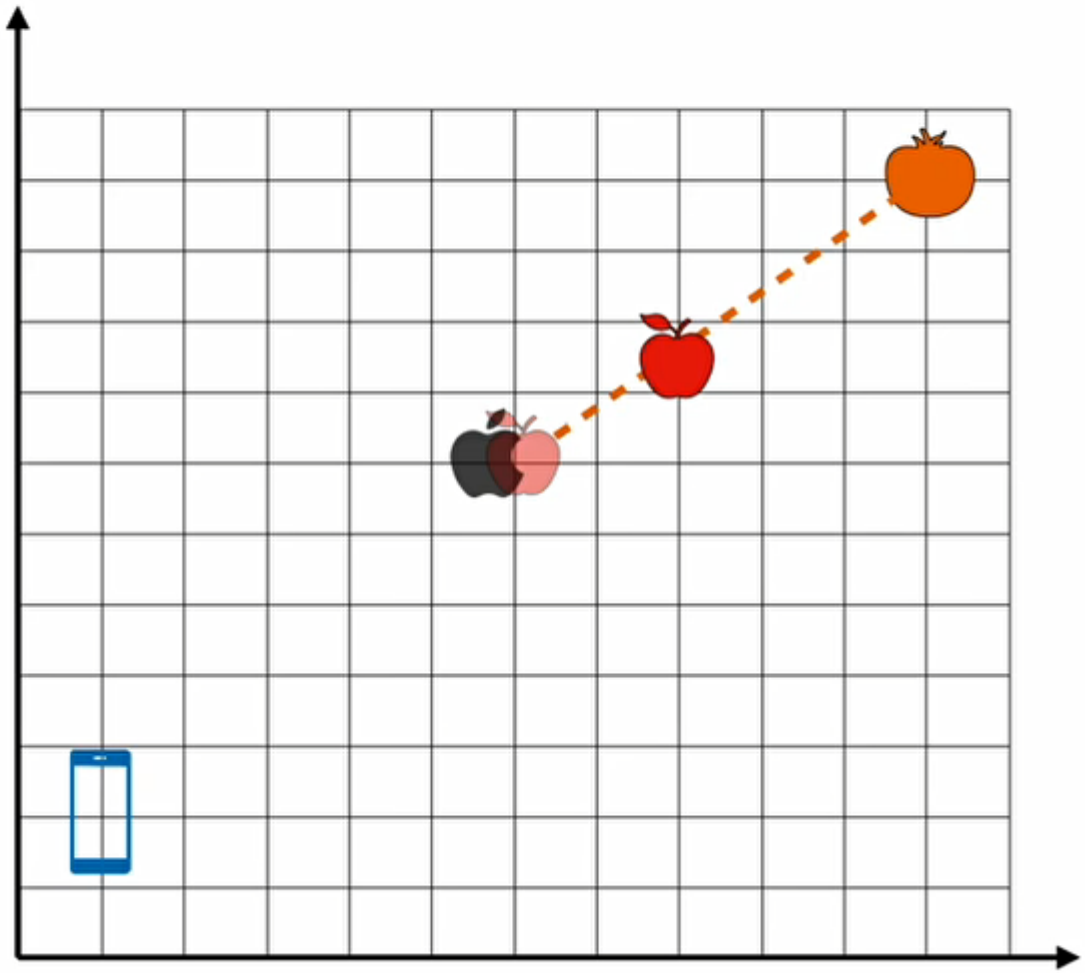


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

WE NEED A CONTEXT, WE NEED TO USE OTHER WORDS IN A SENTENCE TO HELP THE MAIN UNDERSTANDING

Attention

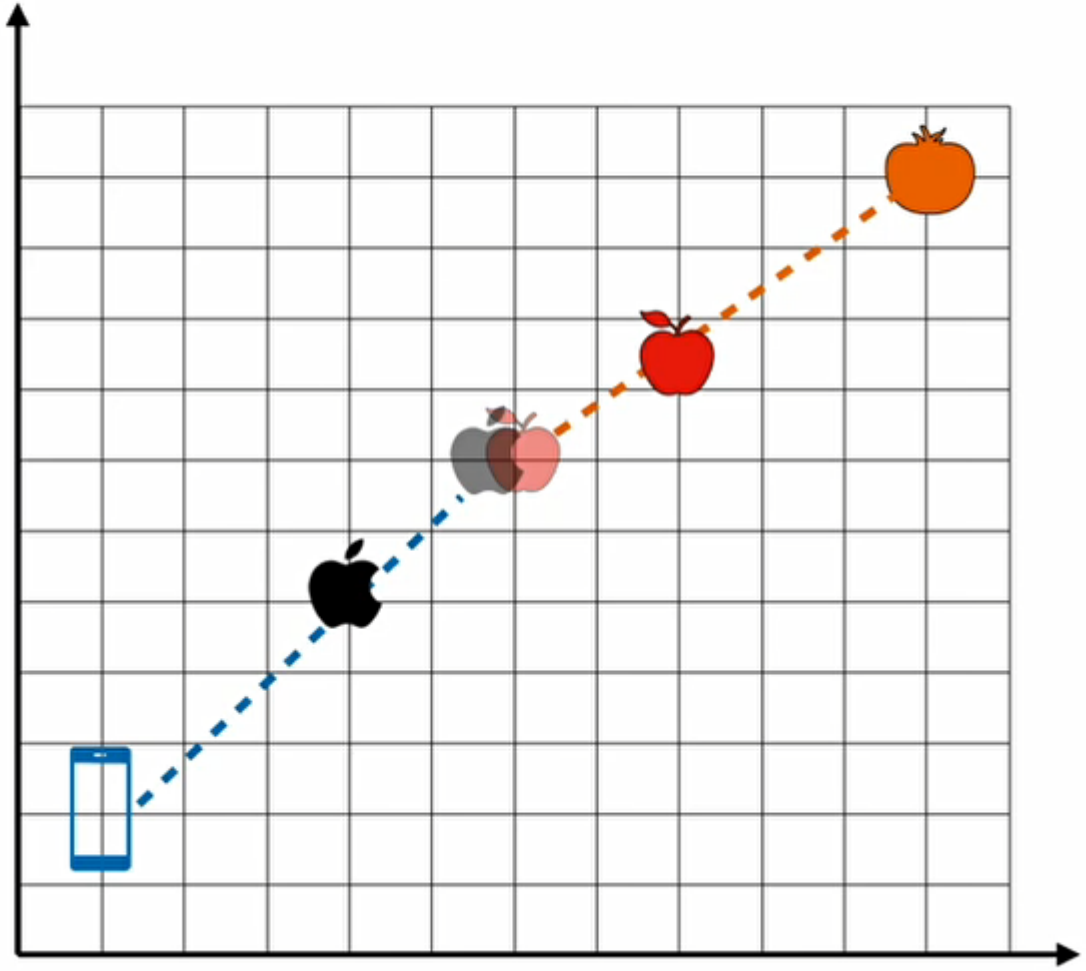


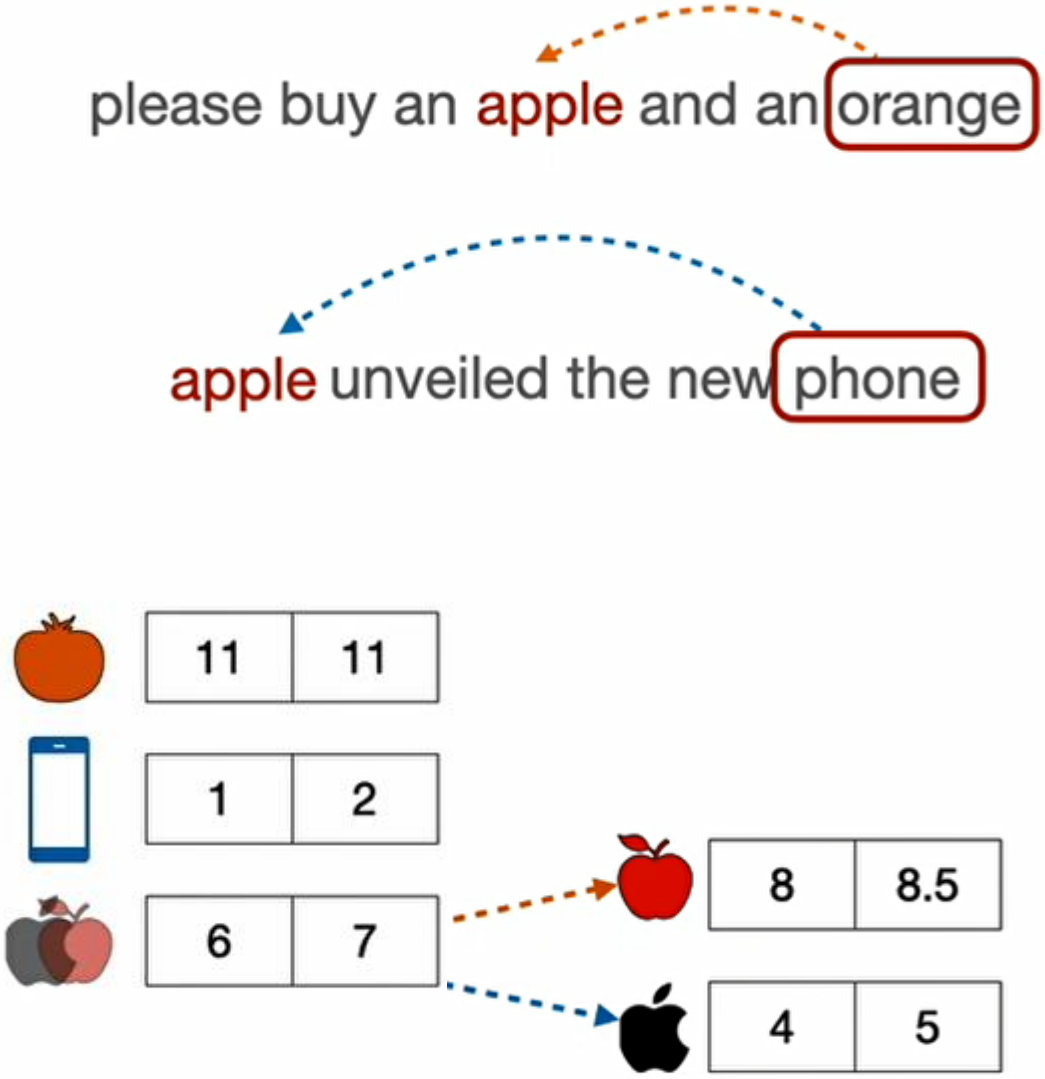
please buy an apple and an orange

apple unveiled the new phone

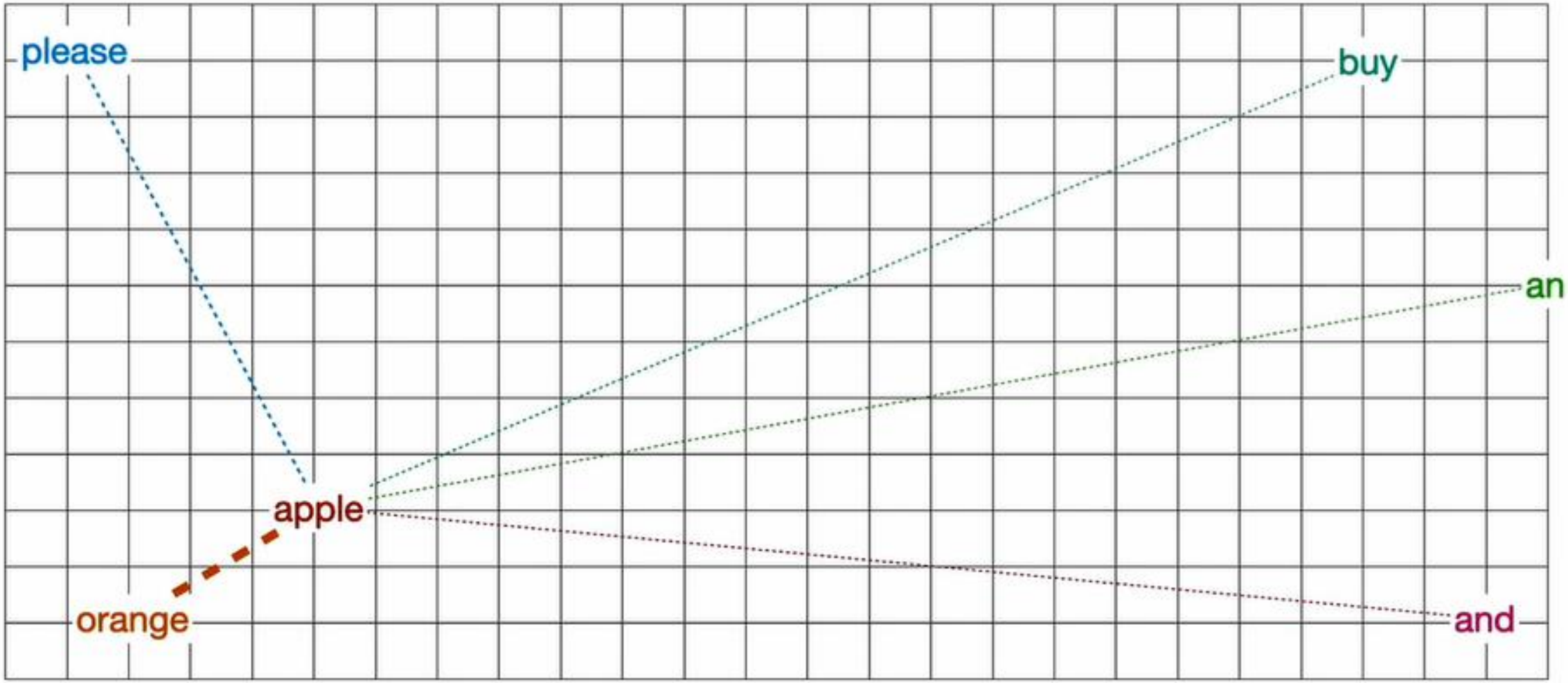
Attention

OF COURSE THIS IS REPEATED SO MANY MANY TIMES !!!



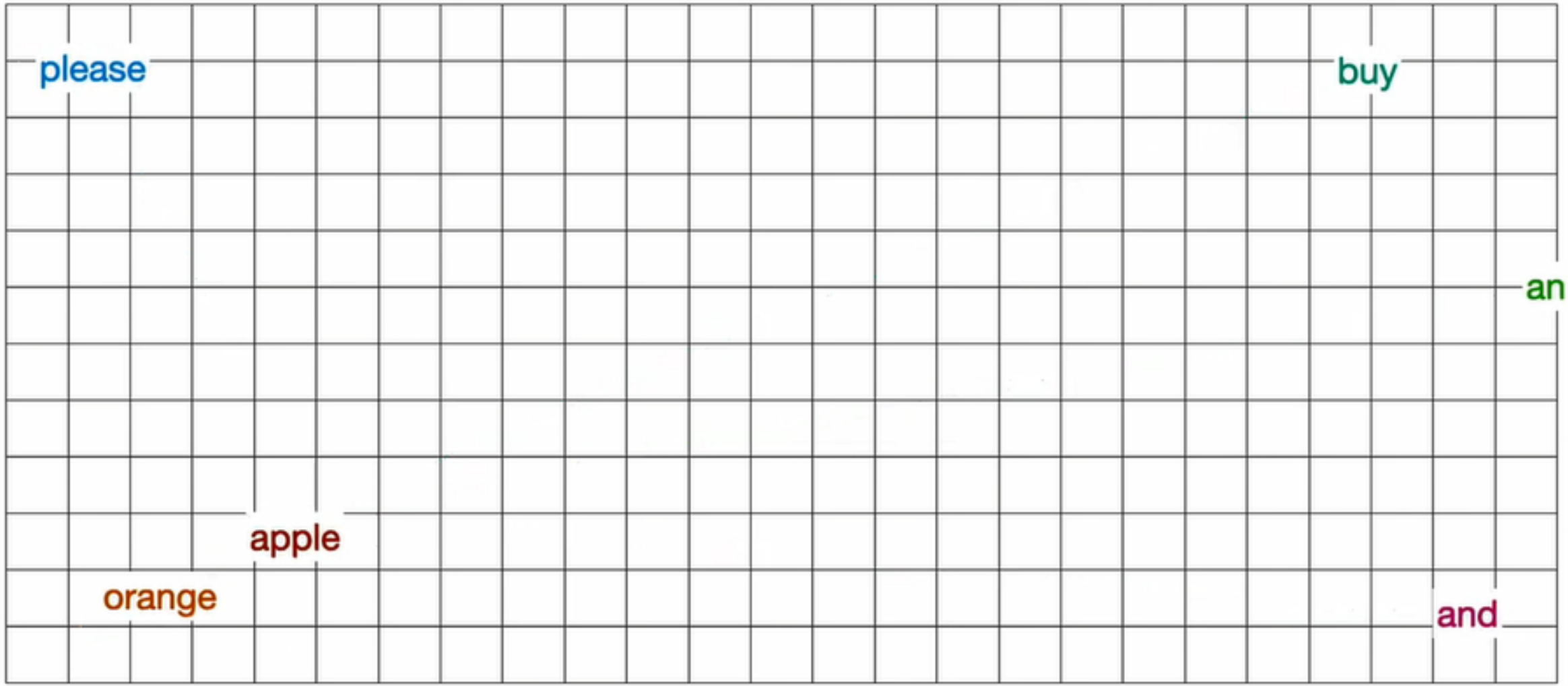


What about the other words?

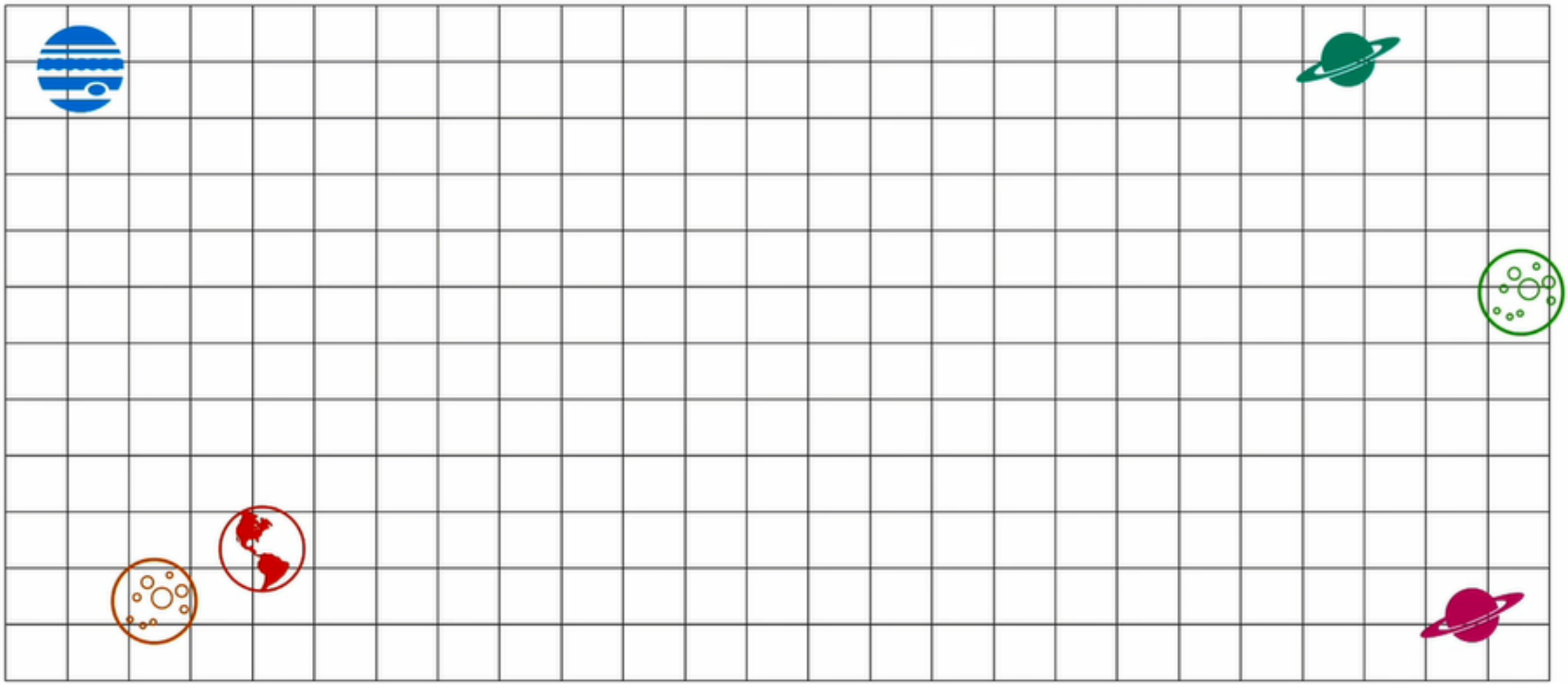


please buy an apple and an orange

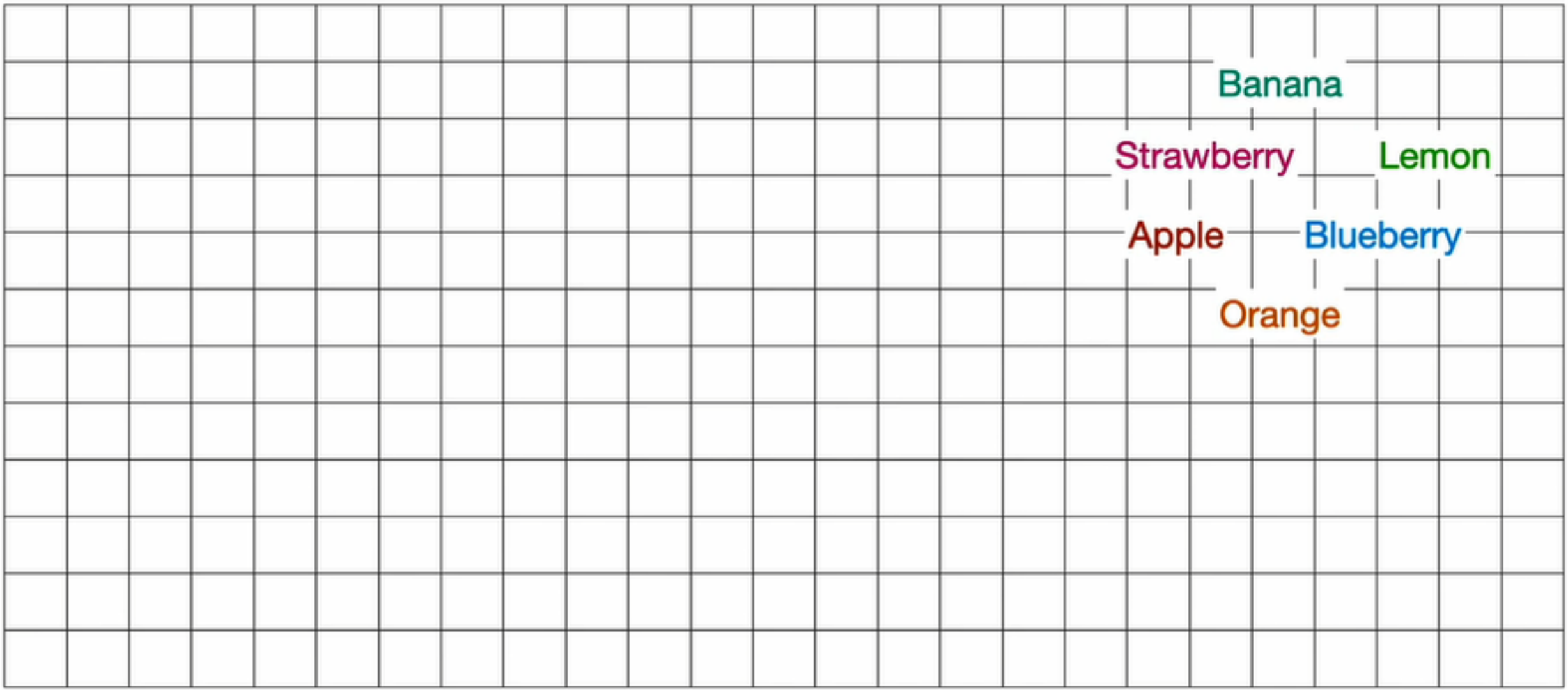
You apply attention to all the words



It’s kind of like gravity…

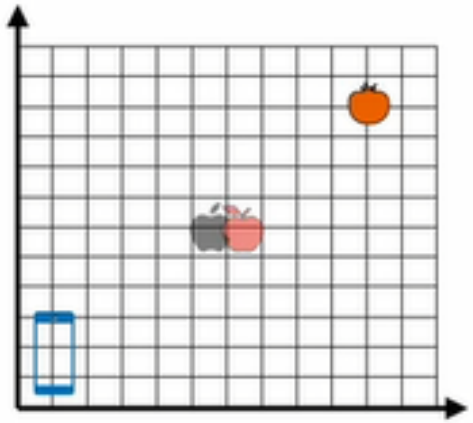


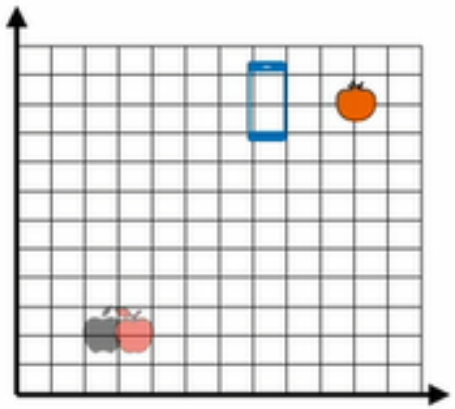
Context pulls

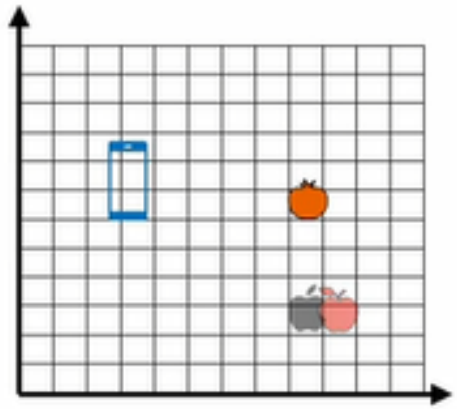


Multi-head attention

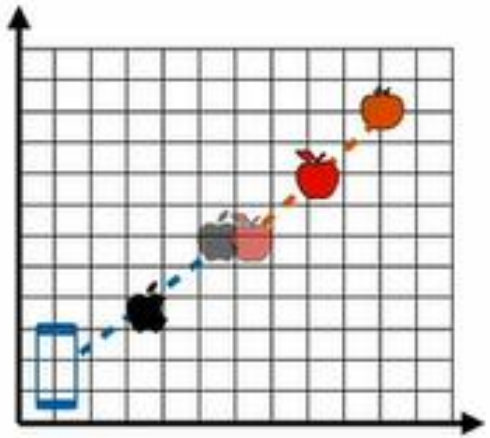
Ideally, we'd like to have lots of embeddings



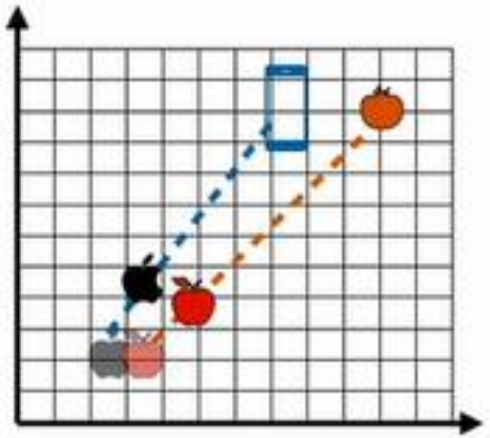




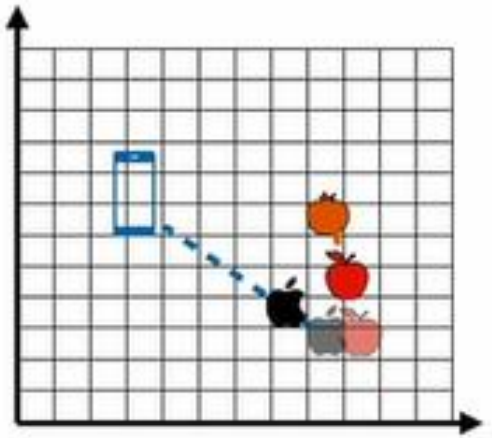
Ideally, we'd like to have lots of embeddings



Good



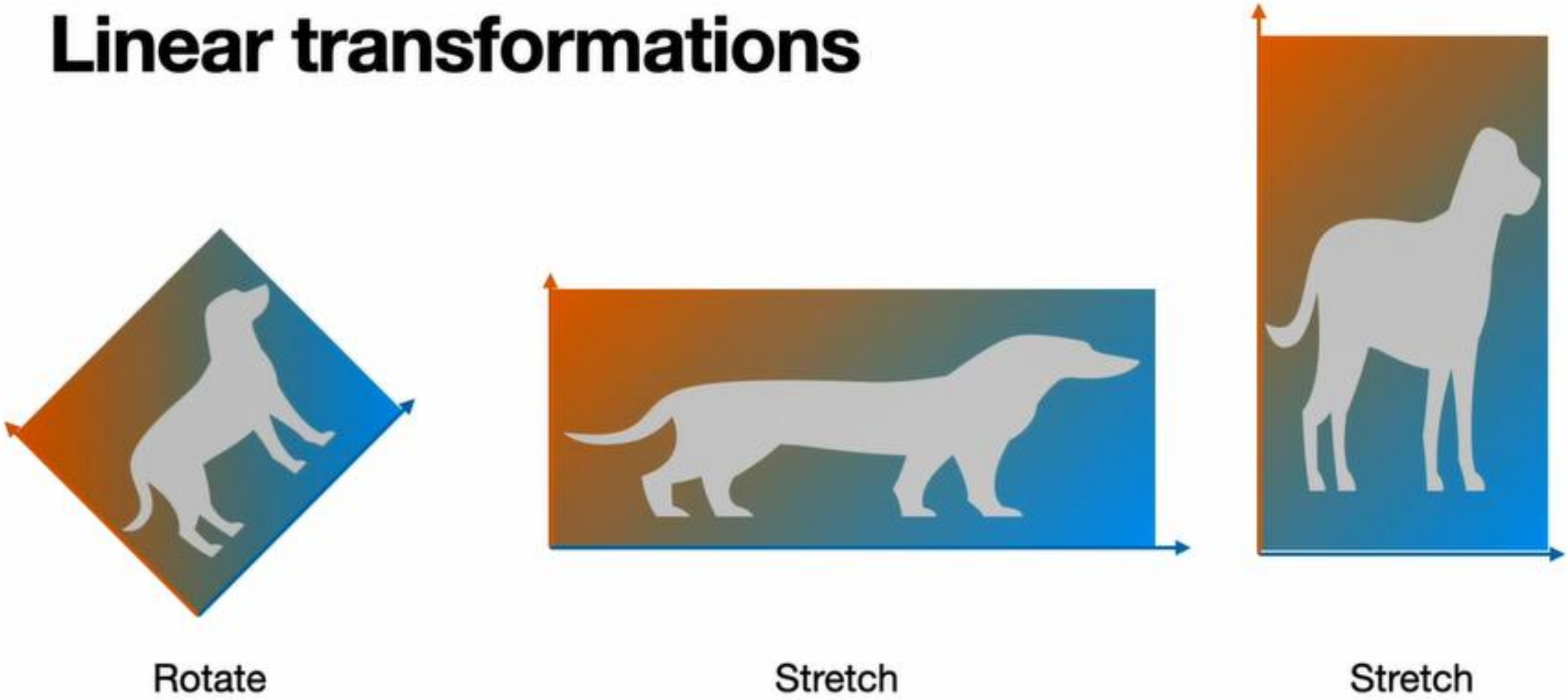
Bad



So-so

Solution: We'll build embeddings by modifying existing embeddings

Linear transformations



Rotate

Stretch

Stretch

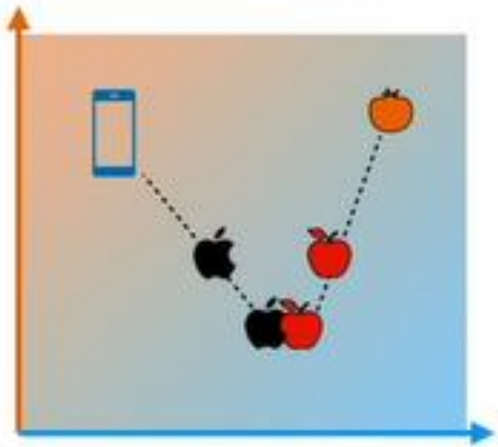
Linear transformation preserves the vector structure of its arguments, as it conserves vector addition and scalar multiplication.

For example, a rotation, a stretching, a shear, and a projection are all examples of linear transformations if they satisfy the properties described above

Linear transformations

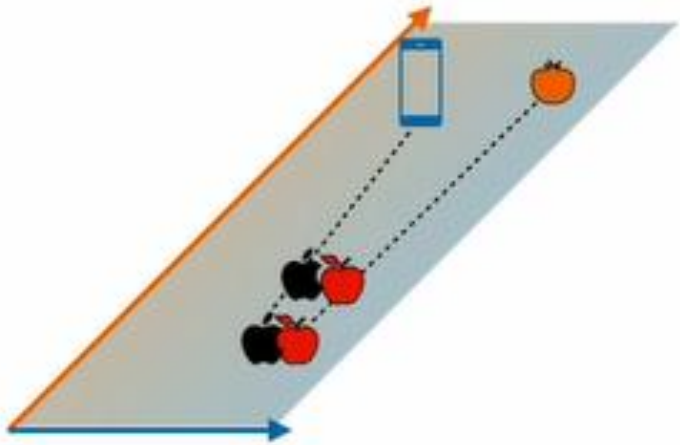
Combination

Get new embeddings from existing ones



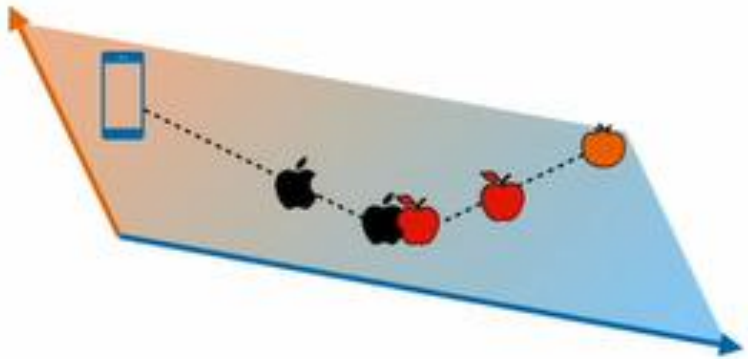
Okay

Score: 1



Bad

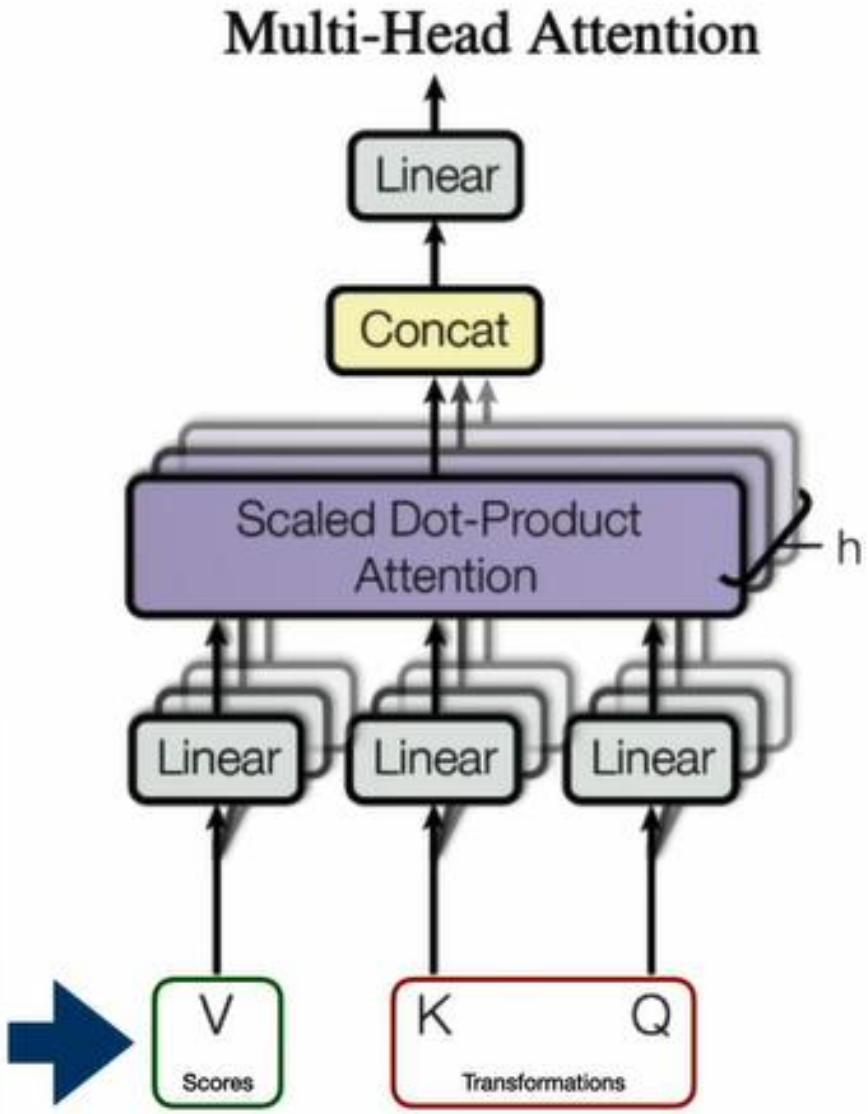
Score: 0.1

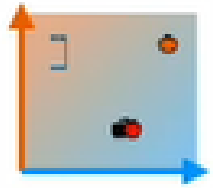


Good

Score: 4

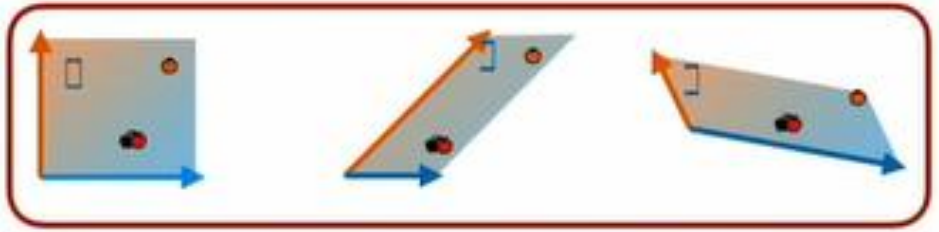
WITH THE TRAIN OF NEAURAL NETWORK, WE TRAIN THE EMBEDDINGS SCORE, SO HERE THE “Q” AND “K” MATRIX COME IN:



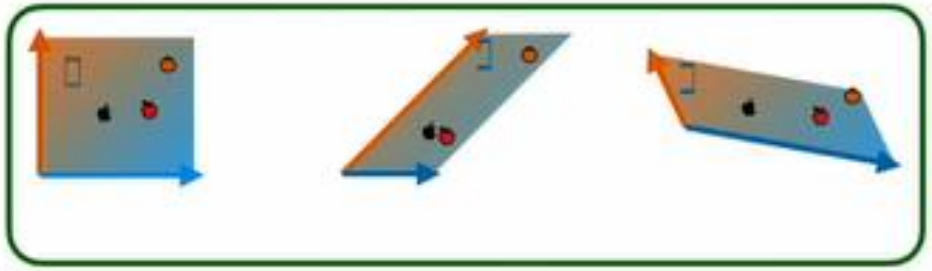


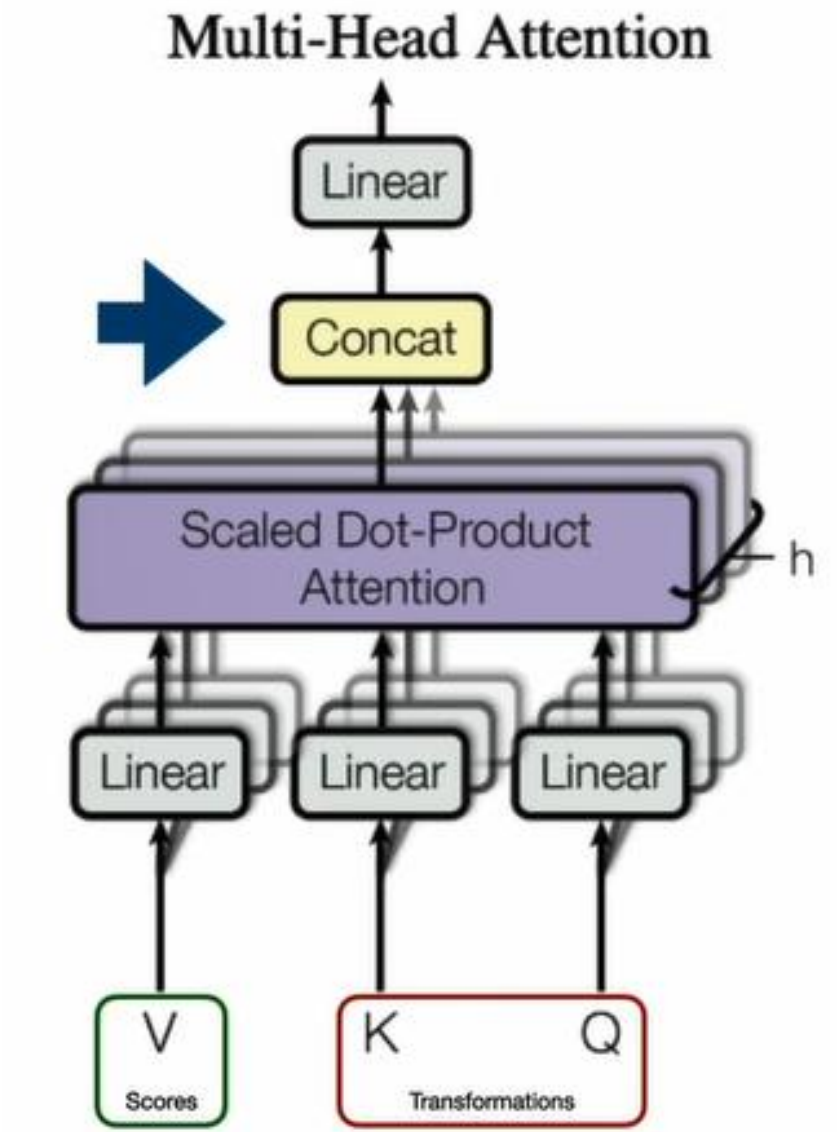
please buy an apple and an orange
apple unveiled the new phone

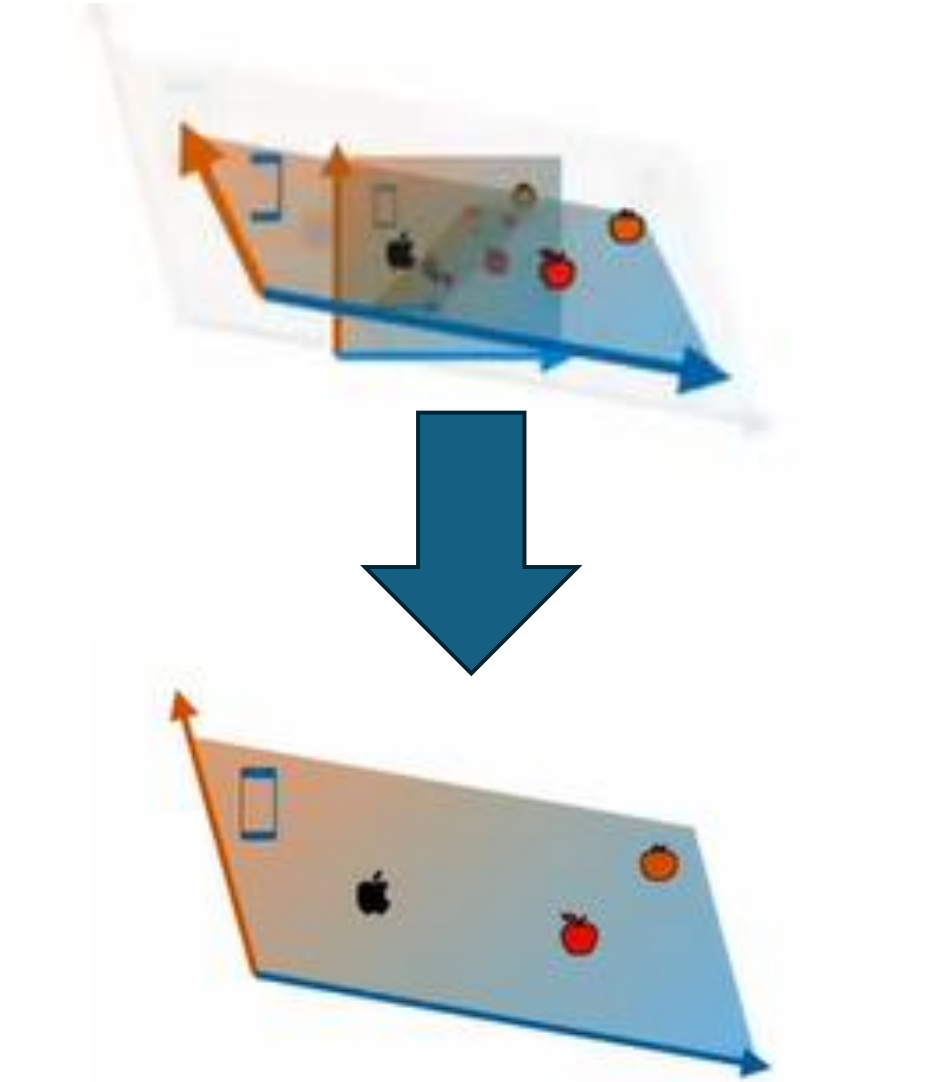
K and Q matrix help the transformationto enanche the similarity



V Matrix help to visualize get better Embeddings

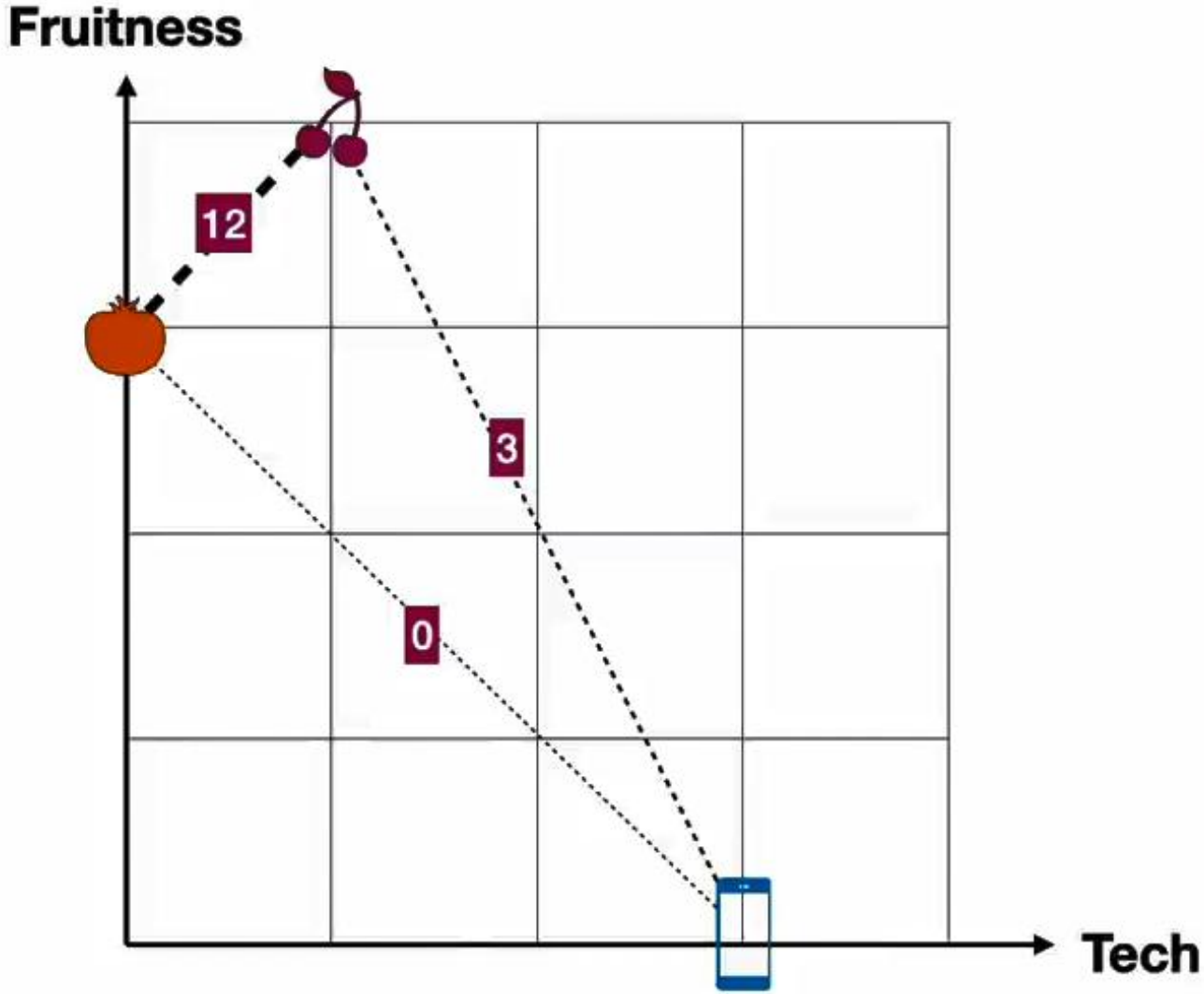



Multi-Head Attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
Scores
K
Q
Transformations



Similarity

Measure 1: Dot product



	Tech	Fruitness
Sim	1	4
	0	3

$$1 \cdot 0 + 4 \cdot 3 = 12$$

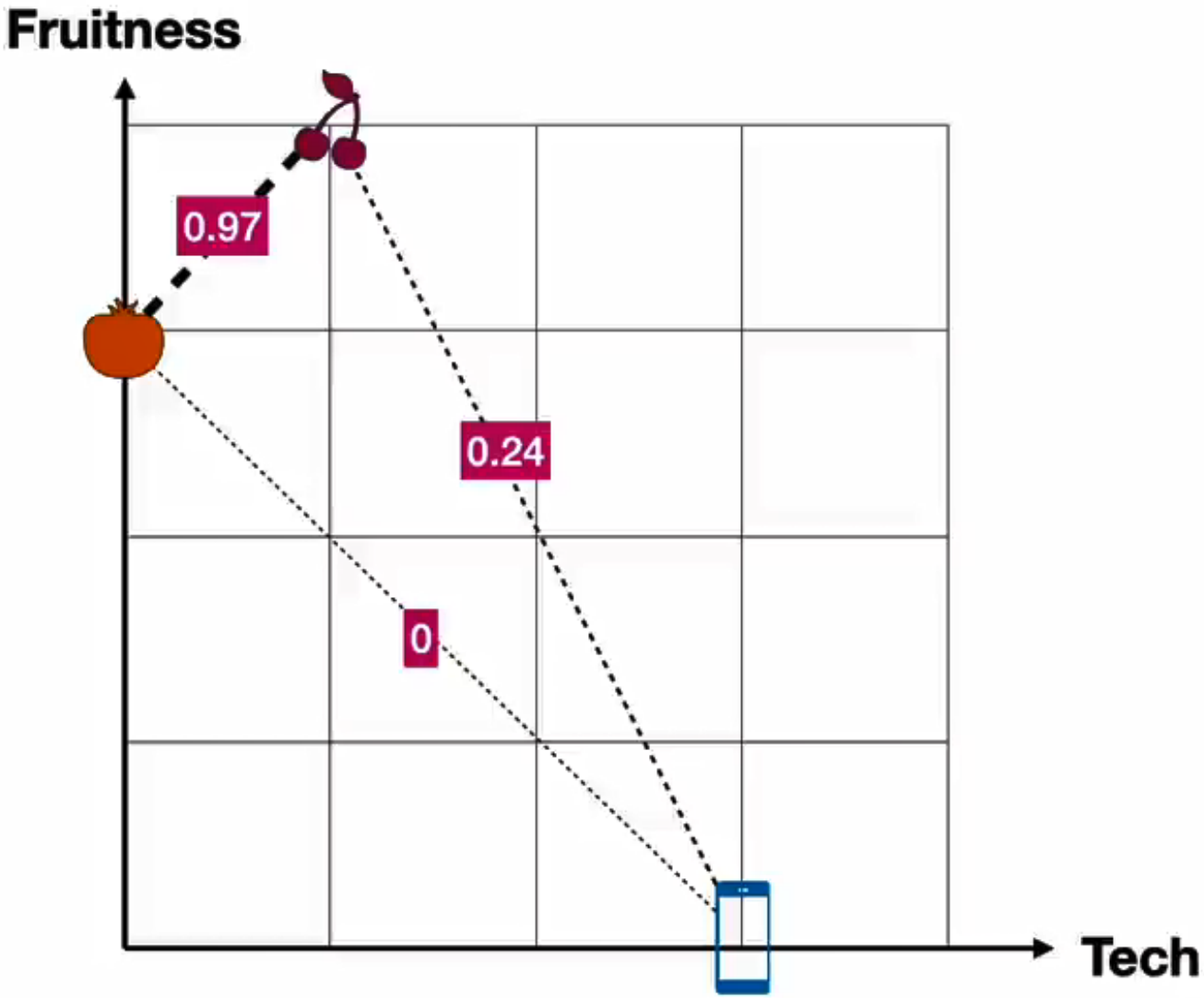
	Tech	Fruitness
Sim	1	4
	3	0

$$1 \cdot 3 + 4 \cdot 0 = 3$$

	Tech	Fruitness
Sim	0	3
	3	0

$$0 \cdot 3 + 3 \cdot 0 = 0$$

Measure 2: Cosine similarity



Sim

$cos(14°) = 0.97$

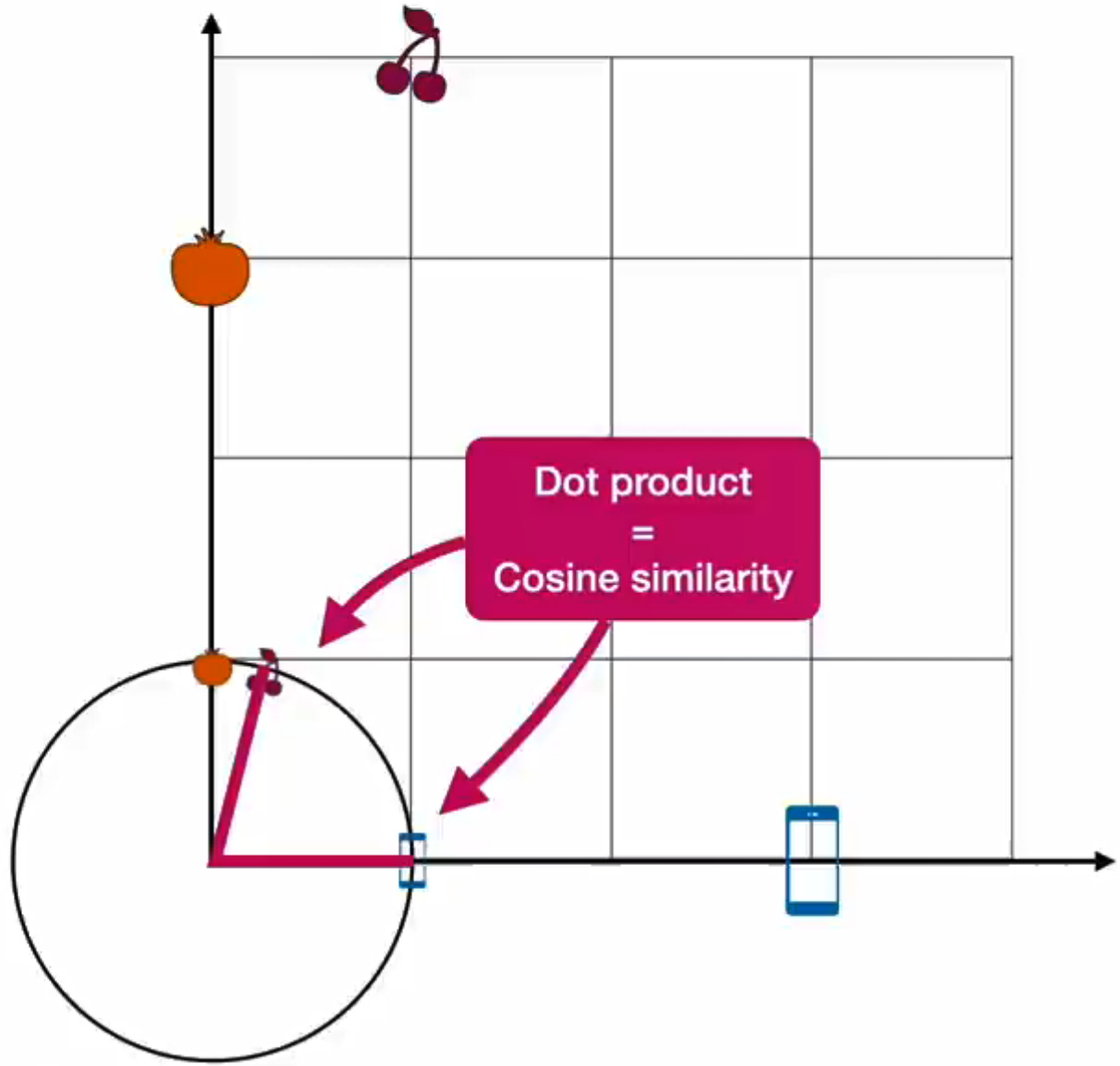
Sim

$cos(76°) = 0.24$

Sim

$cos(90°) = 0$

Dot product and cosine similarity



Measure 3: Scaled dot product

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

Sim

1	4

0	3

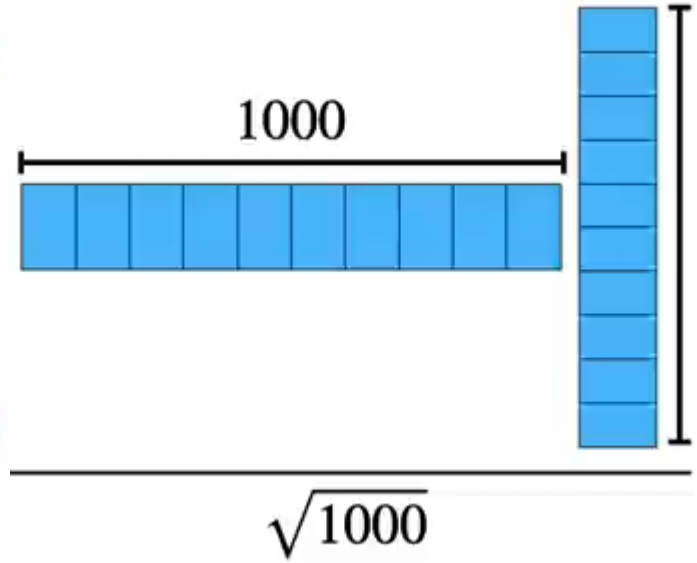
$$1 \cdot 0 + 4 \cdot 3 = 12 \longrightarrow \frac{12}{\sqrt{2}} = 8.49$$

Sim

1	4

3	0

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



Sim

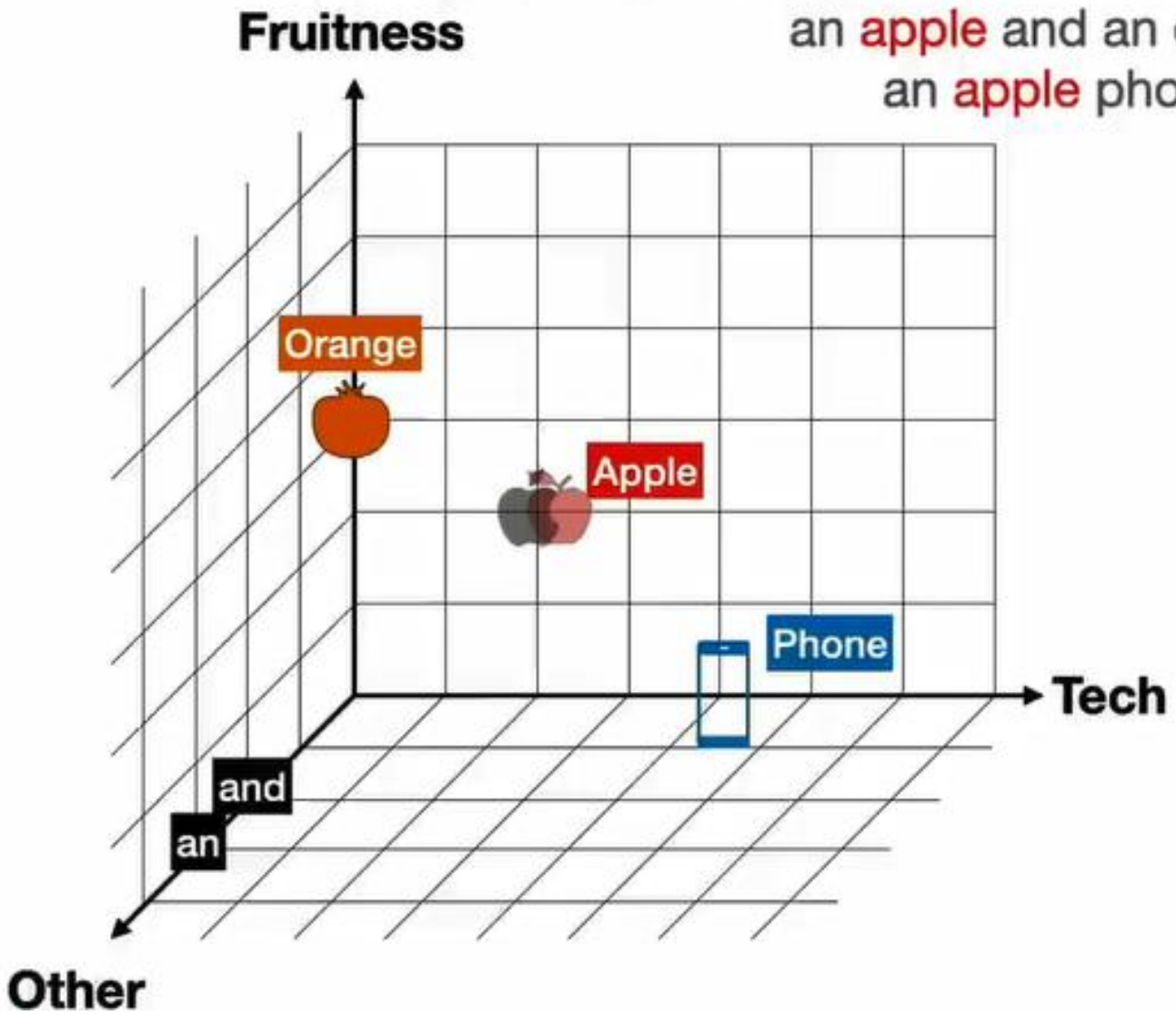
0	3

3	0

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

NOW, LET'S GET TO ATTENTiON!

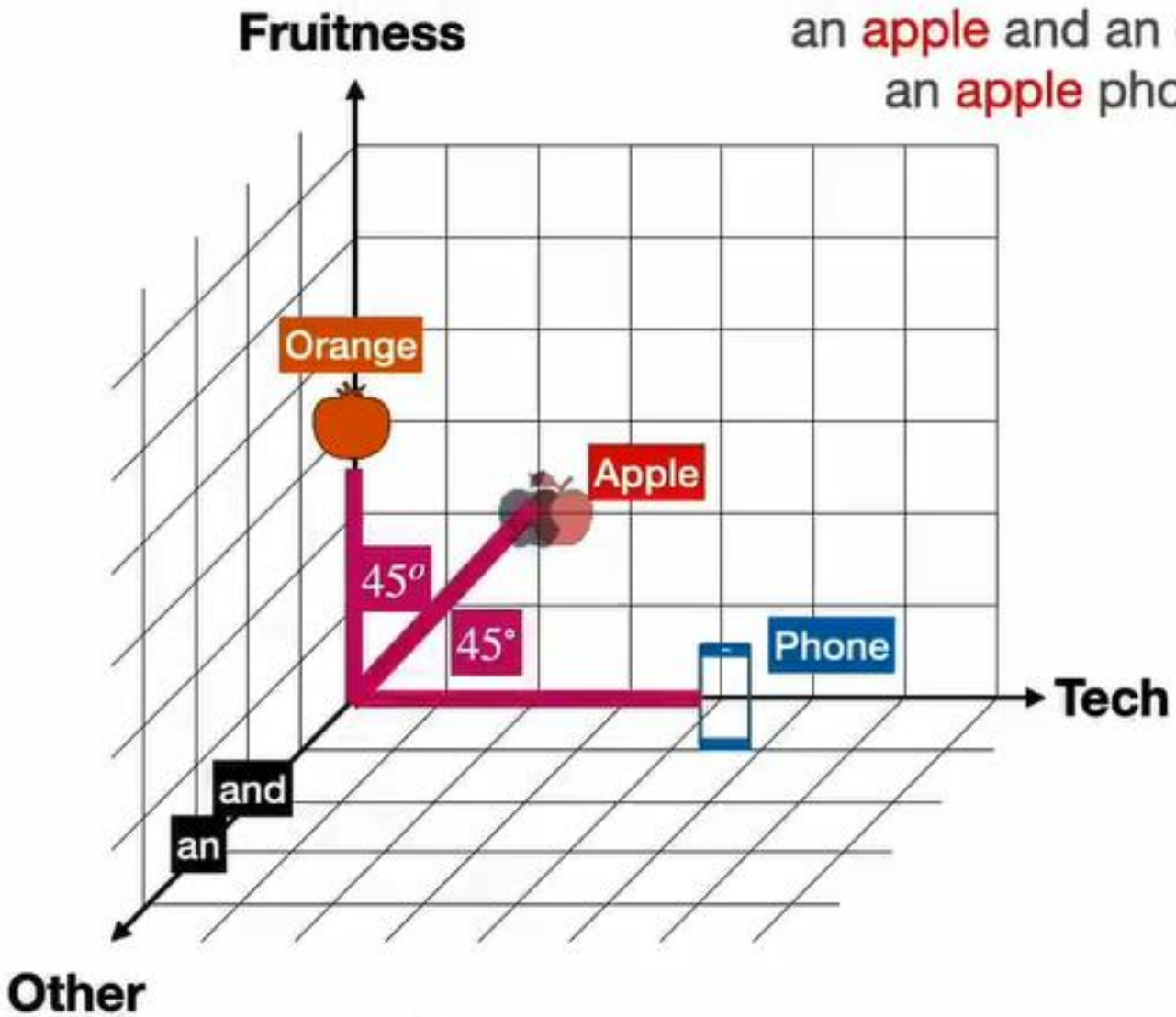
Cosine similarity



	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				
Phone		1			
Apple			1		
And				1	
An					1

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	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

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 → 1 Orange + 0.71 Apple

Apple → 0.71 Orange + 1 Apple

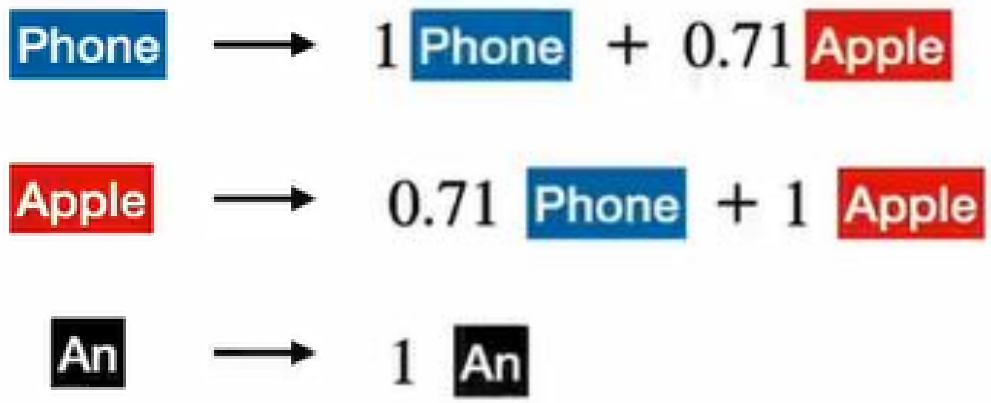
And → 1 And + 1 An

An → 1 An + 1 And

Word math

an apple phone

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



Normalization

Want coefficients to add to 1

Orange $\longrightarrow \dfrac{1\ \text{Orange} + 0.71\ \text{Apple}}{1 + 0.71} = 0.58\ \text{Orange} + 0.42\ \text{Apple}$

Solution?

$$x \longmapsto e^x$$

Softmax

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

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

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

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 → 0.57 Orange + 0.43 Apple

Apple → 0.43 Orange + 0.57 Apple

And → 0.5 And + 0.5 An

An → 0.5 An + 0.5 And

an apple phone

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

Phone → 0.57 Phone + 0.43 Apple

Apple → 0.43 Phone + 0.57 Apple

An → 1 An

an apple and an orange

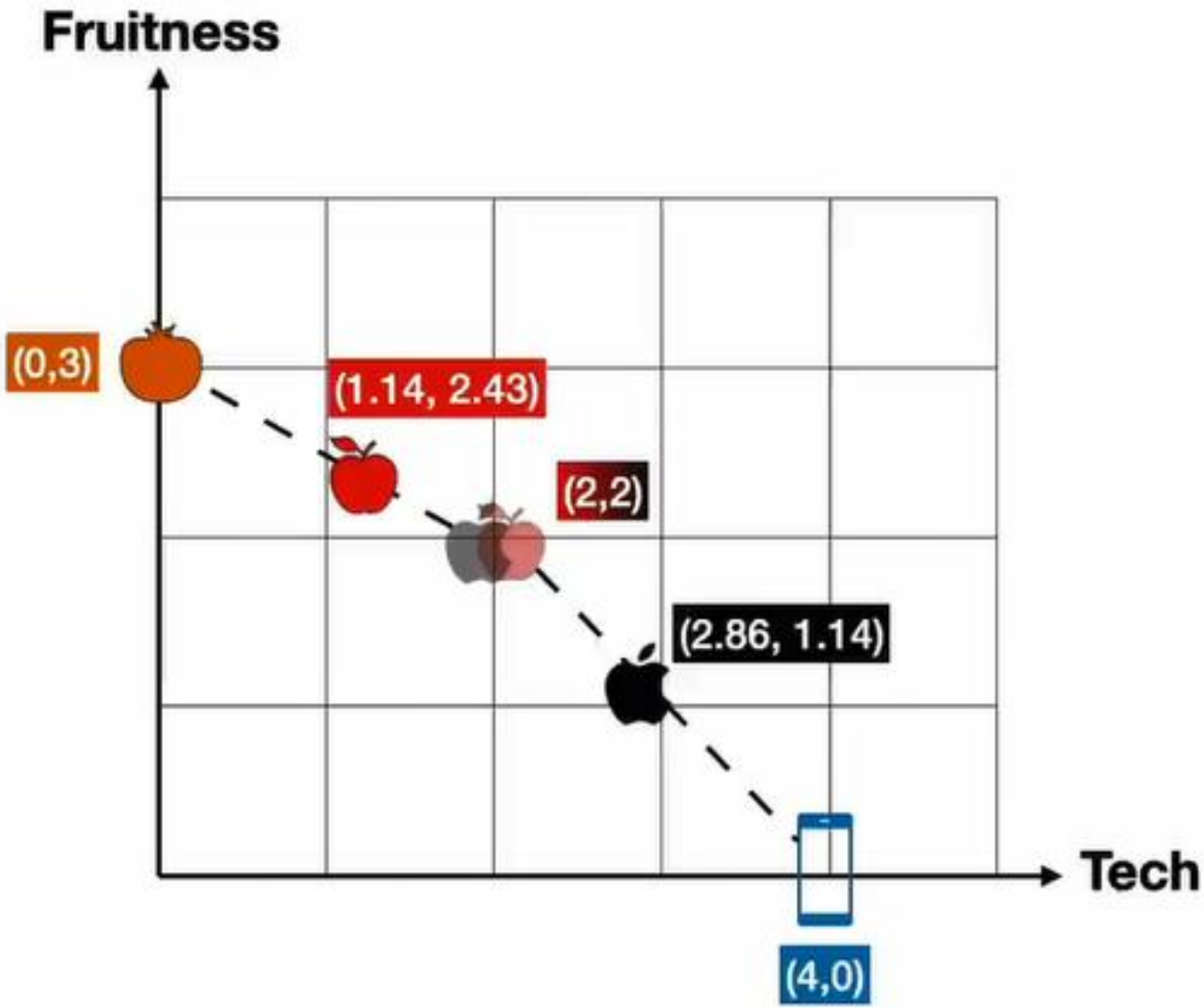
	Orange	Apple	And	An
Orange	1	0.71	0	0

$$\frac{e^{1}\,\text{Orange} + e^{0.71}\,\text{Apple} + e^{0}\,\text{And} + e^{0}\,\text{An}}{e^{1} + e^{0.71} + e^{0} + e^{0}}$$

Ok, I'm kinda lying to you…

$$\text{Orange} \longrightarrow 0.57\,\text{Orange} + 0.43\,\text{Apple}$$

$$\text{Orange} \longrightarrow 0.4\,\text{Orange} + 0.3\,\text{Apple} + 0.15\,\text{And} + 0.15\,\text{An}$$



an apple and an orange

Apple → 0.43 Orange + 0.57 Apple

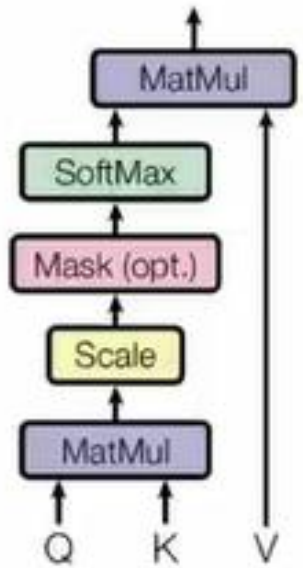
an apple phone

Apple → 0.43 Phone + 0.57 Apple

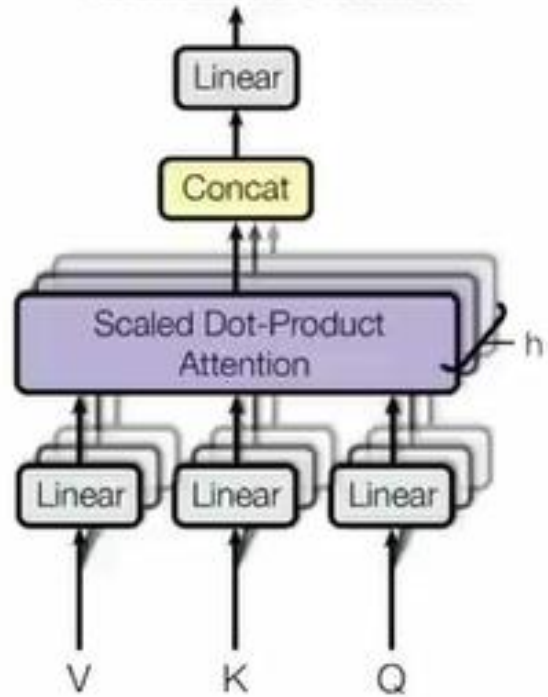
The keys, queries, and values matrices

Attention

Scaled Dot-Product Attention



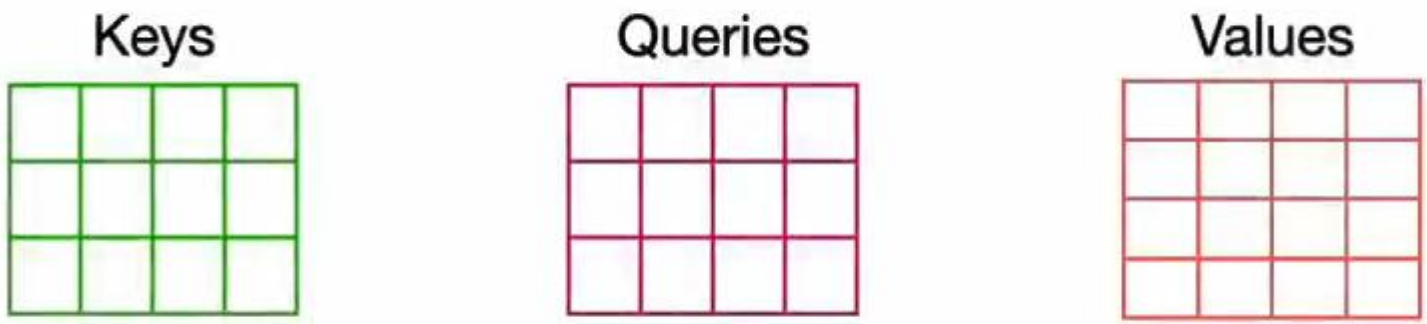
Multi-Head Attention



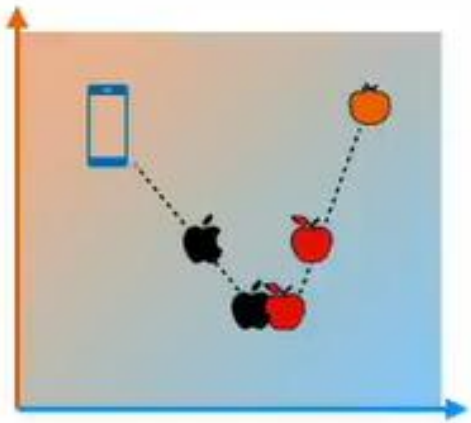
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

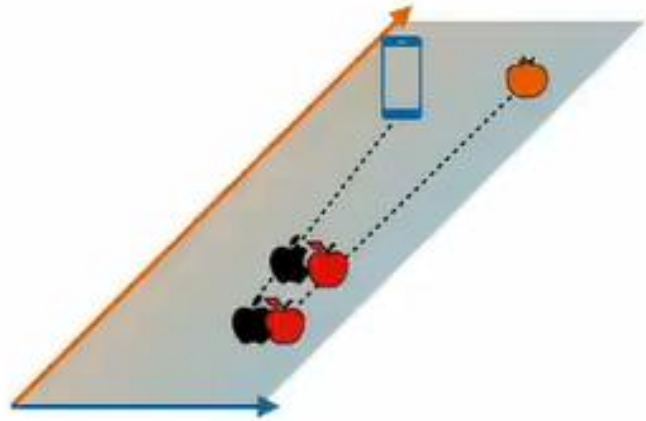
The Key and Query matrices



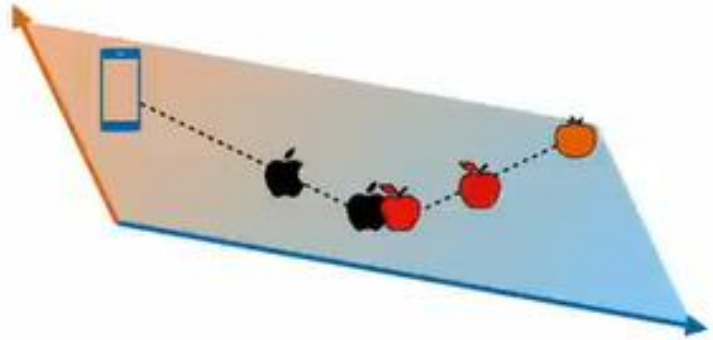
Get new embeddings from existing ones



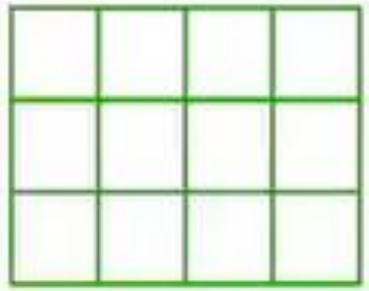
Okay

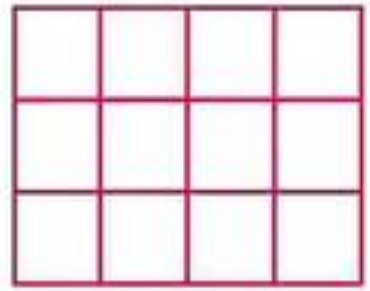


Bad

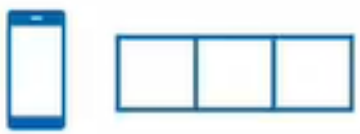


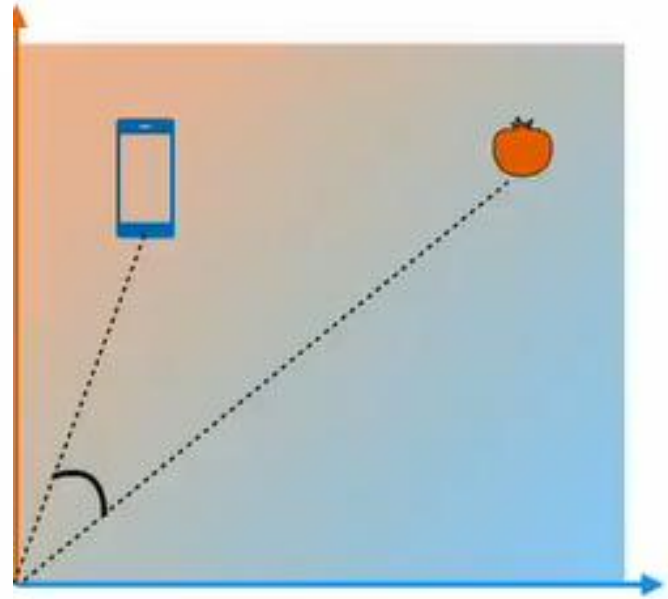
Good





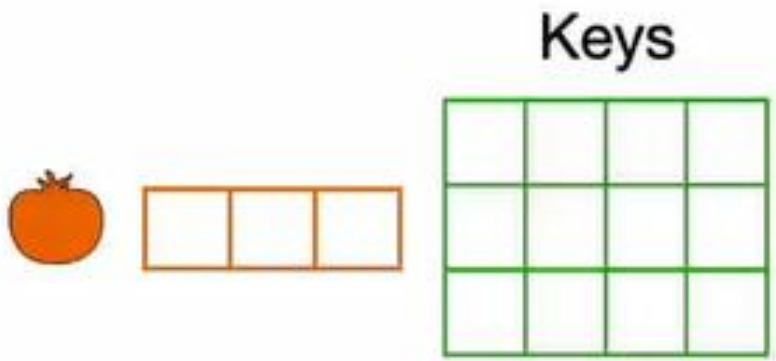
Similarity

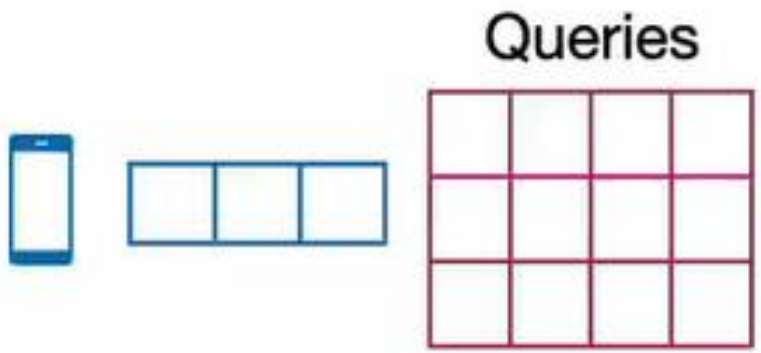


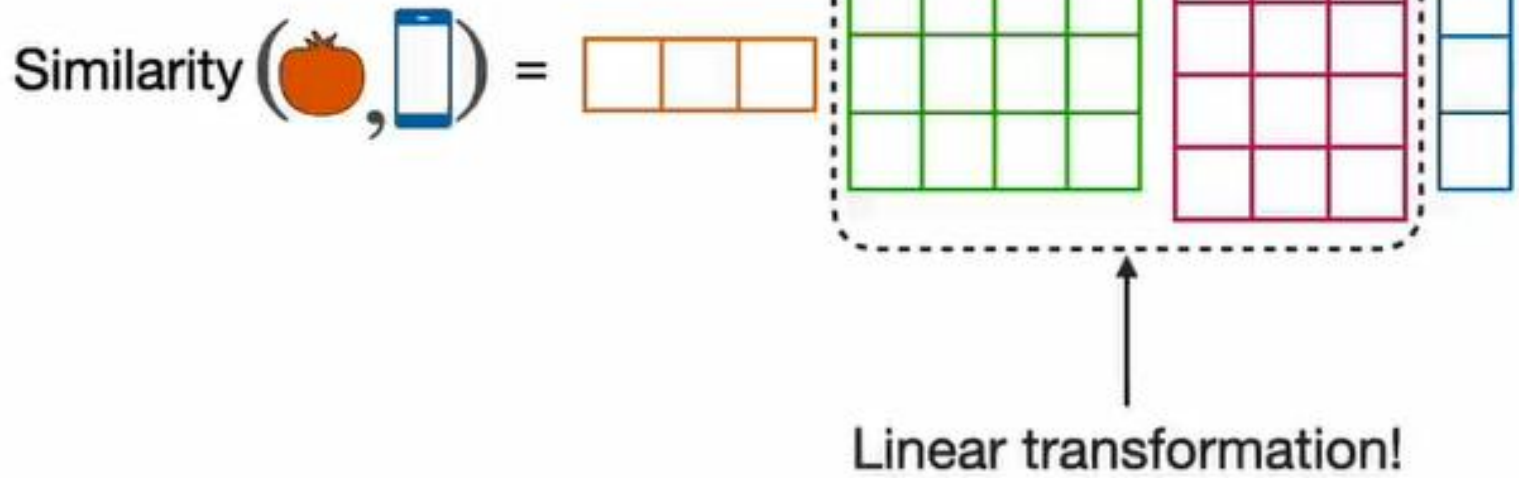


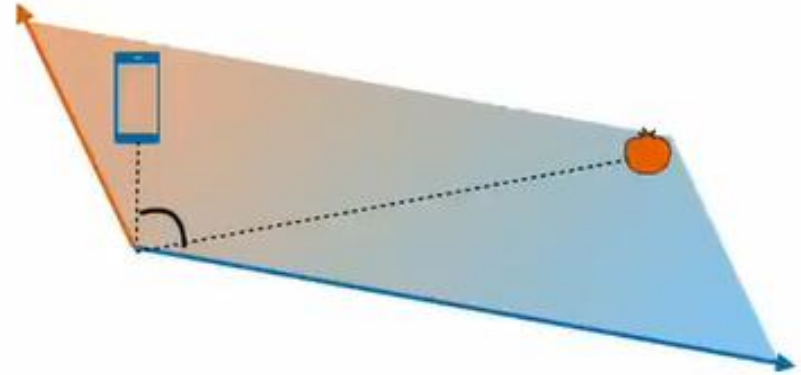
Similarity  = 

Keys and Queries Matrices

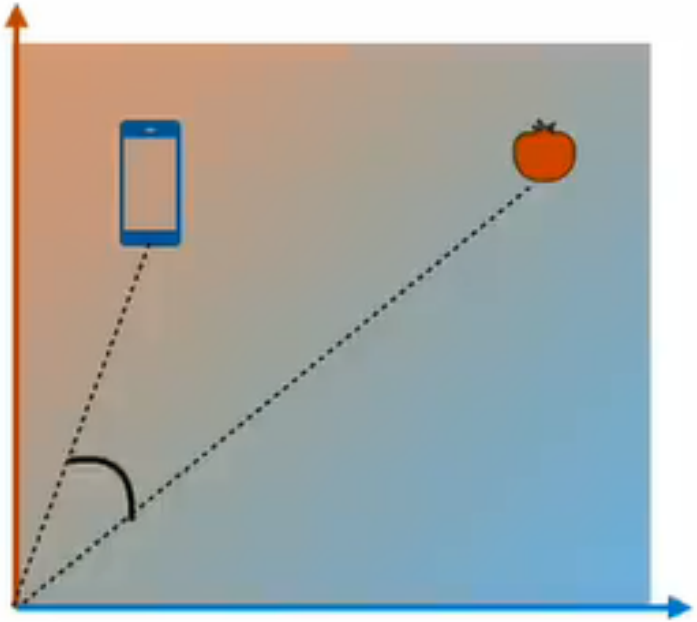


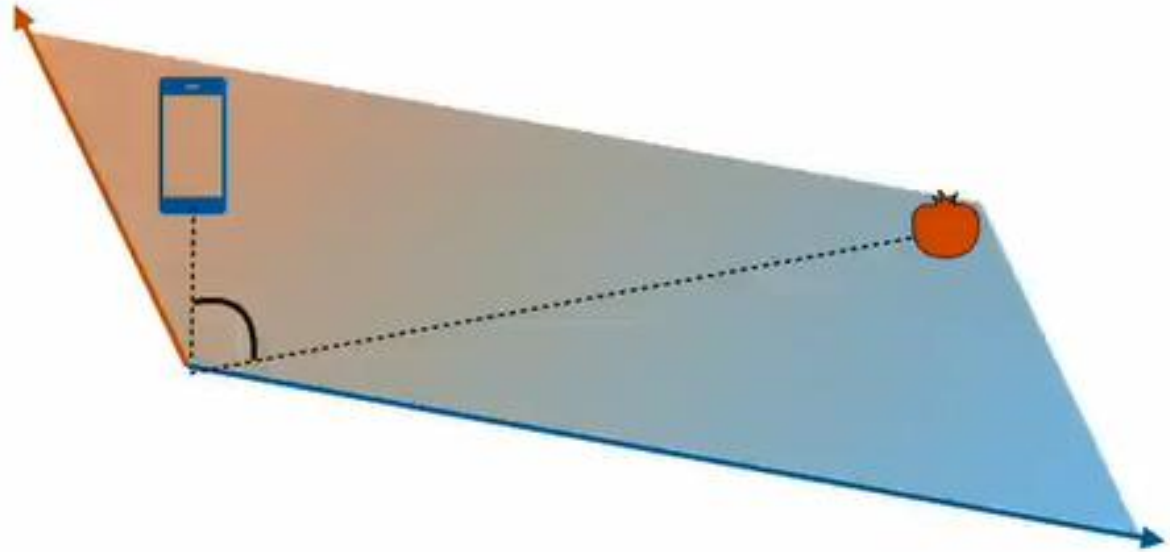


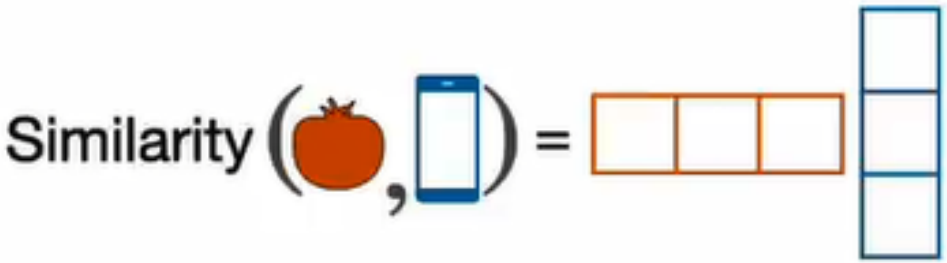


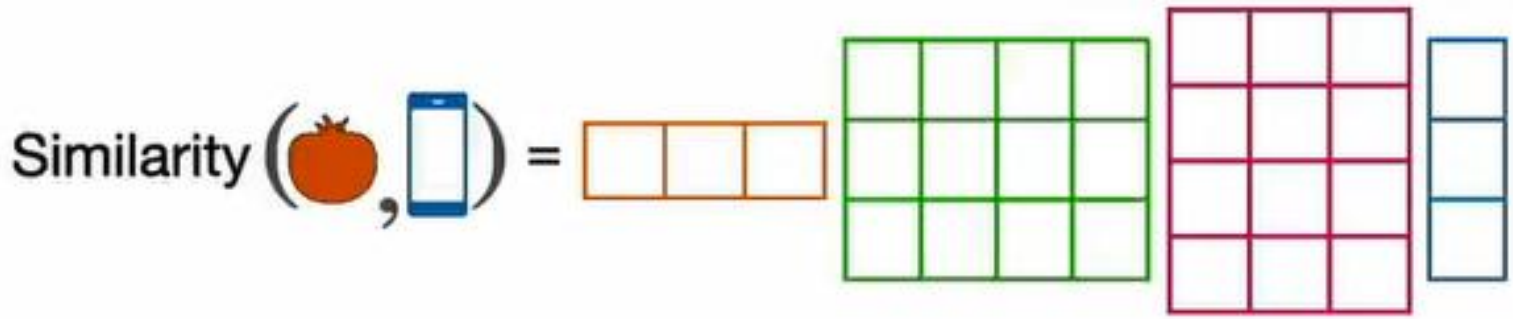


Similarity on a transformed embedding



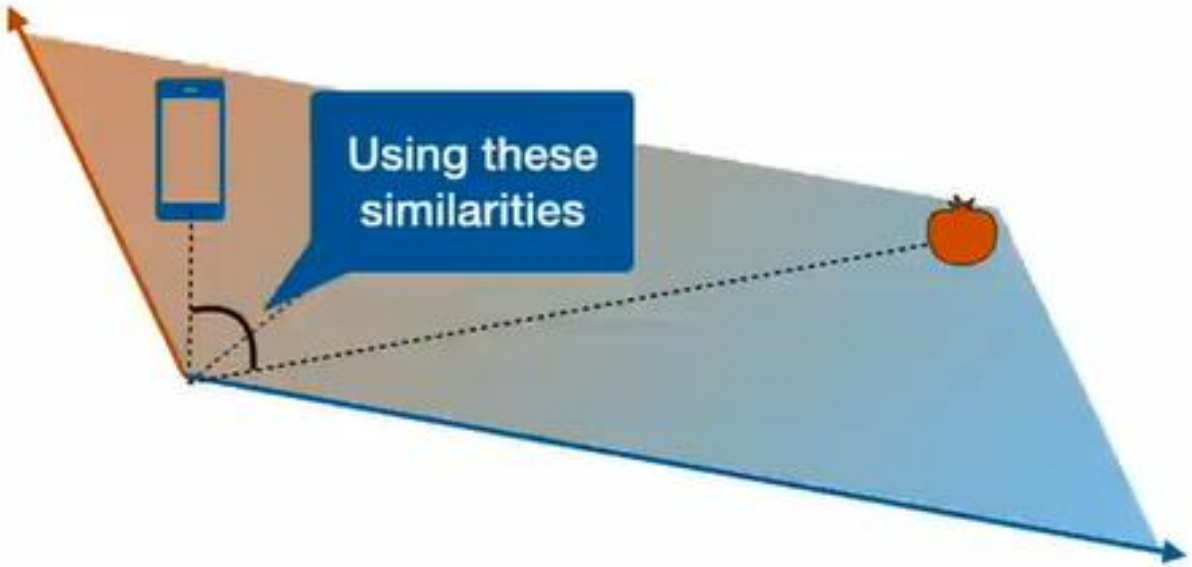






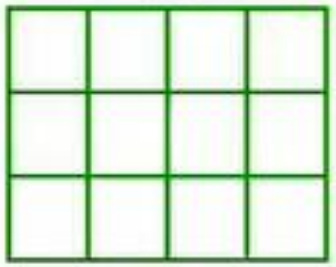
The values matrix

Values matrix

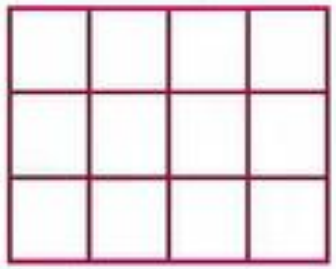


Best embedding for finding similarities

Keys

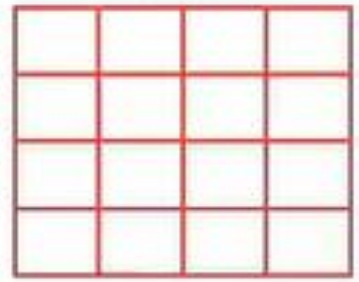


Queries

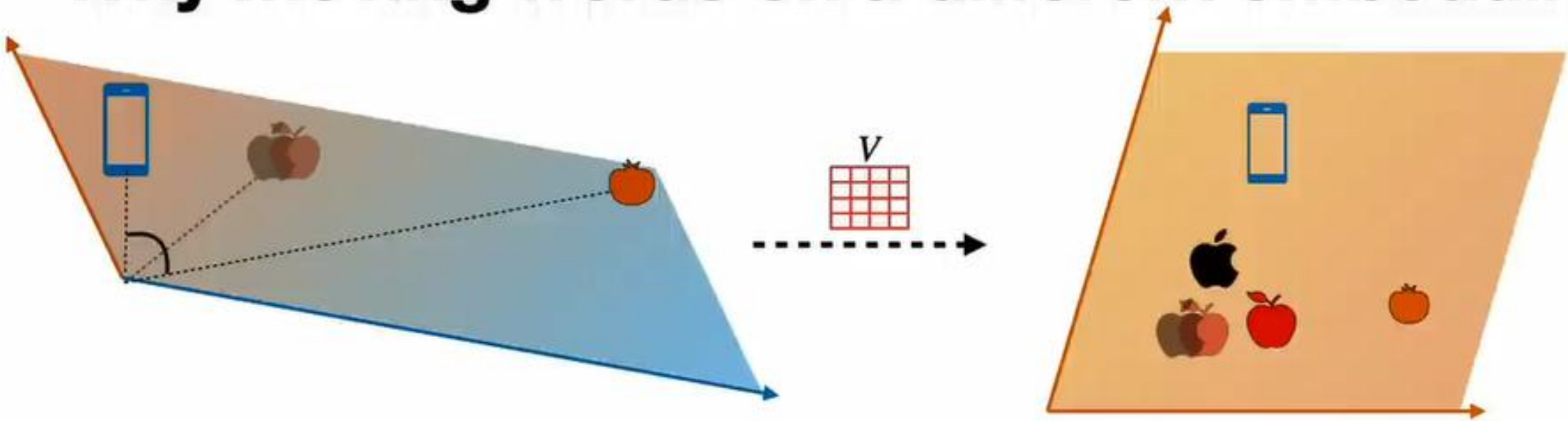


Best embedding for finding the next word

Values



Why moving words on a different embedding?



Best embedding for finding similarities

Best embedding for finding the next word

This embedding(s) know features of the words

- Color
- Size
- Fruitness
- Technology

This embedding knows when two words could appear in the same context

I want to buy a _____

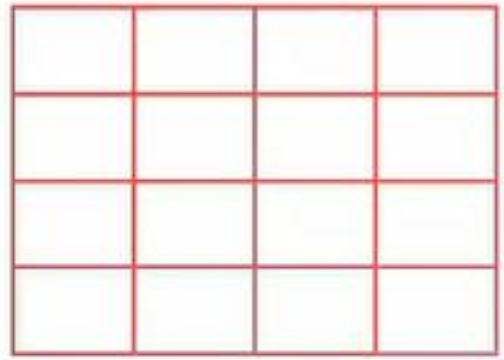
- car
- apple
- phone

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



=

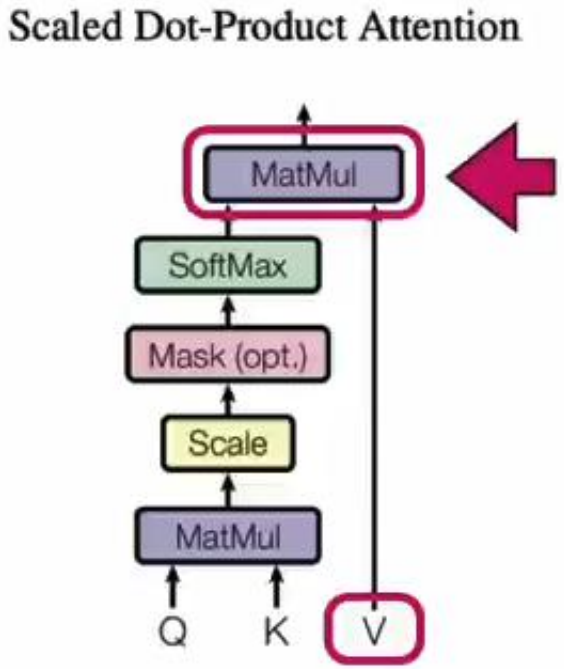
	Orange	Apple	And	An
Orange	v_{11}	v_{12}	v_{13}	v_{14}
Apple	v_{21}	v_{22}	v_{23}	v_{24}
And	v_{31}	v_{32}	v_{33}	v_{34}
An	v_{41}	v_{42}	v_{43}	v_{44}

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

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

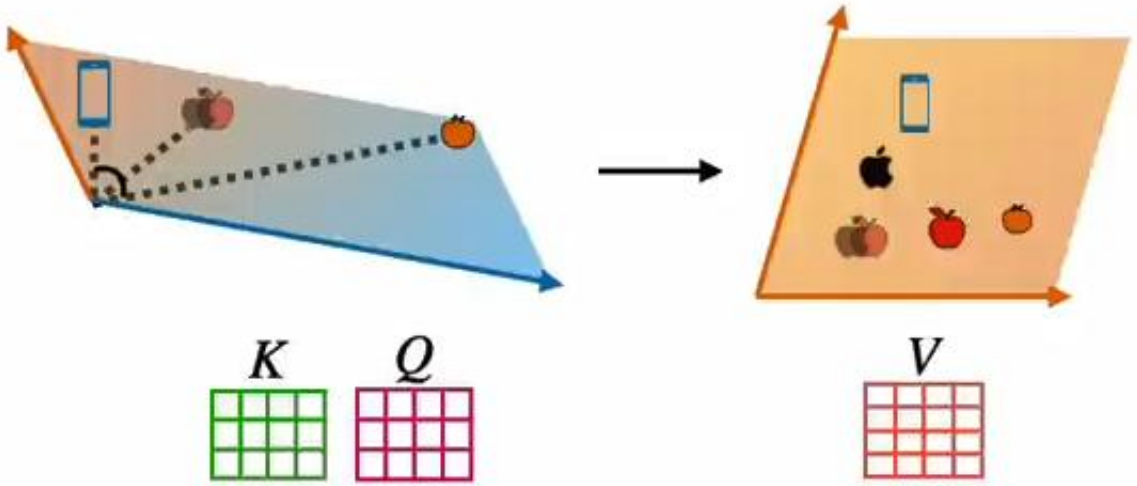
MULTI-HEAD ATTENTION

Self-attention

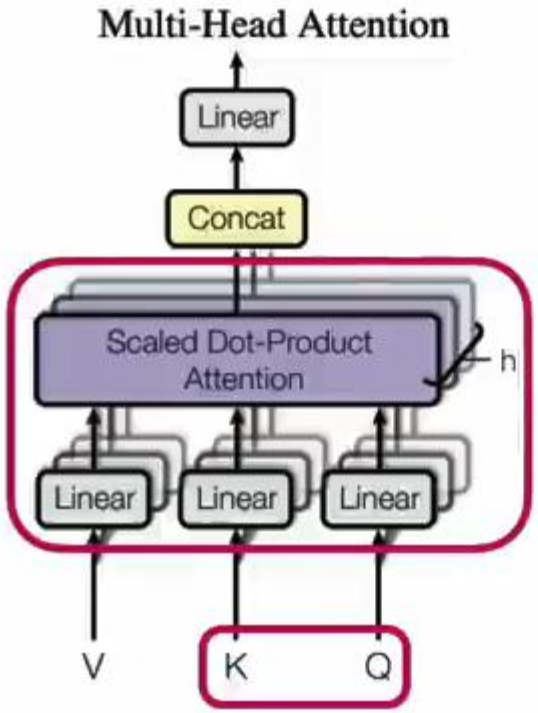


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}V)$$

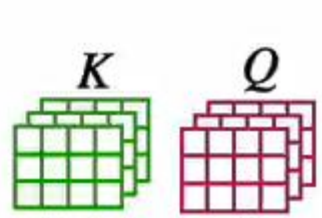
WE WILL USE THE VALUE MATRIX TO TRANSFOM THIS EMBEDDINGS IN A BETTER ONE. SO WE USE THE LINEAR TRANSFORMATION TO FIND THE EMBEDDING ON THE RIGHT AND MOVE THE WORDS IN A BETTER WAY.

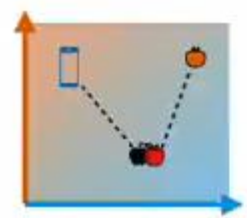


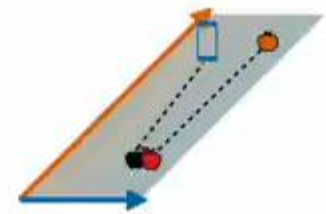
Multi-head attention

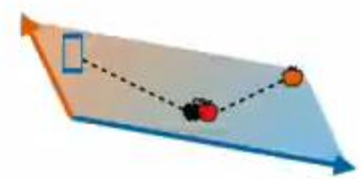


$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

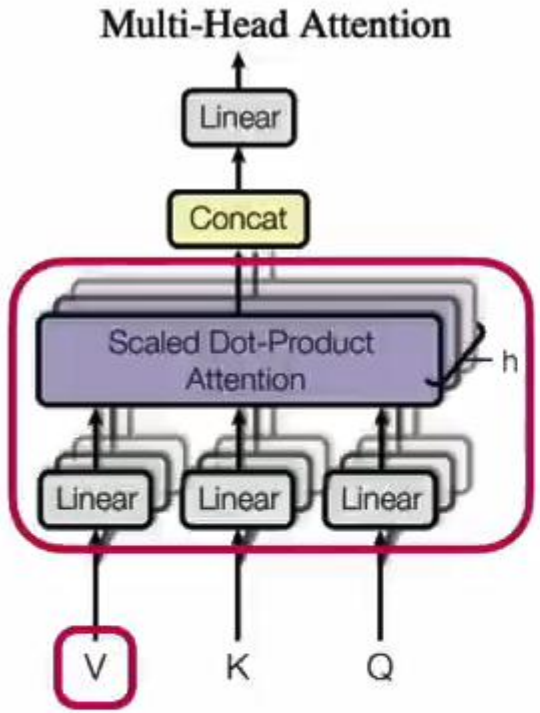






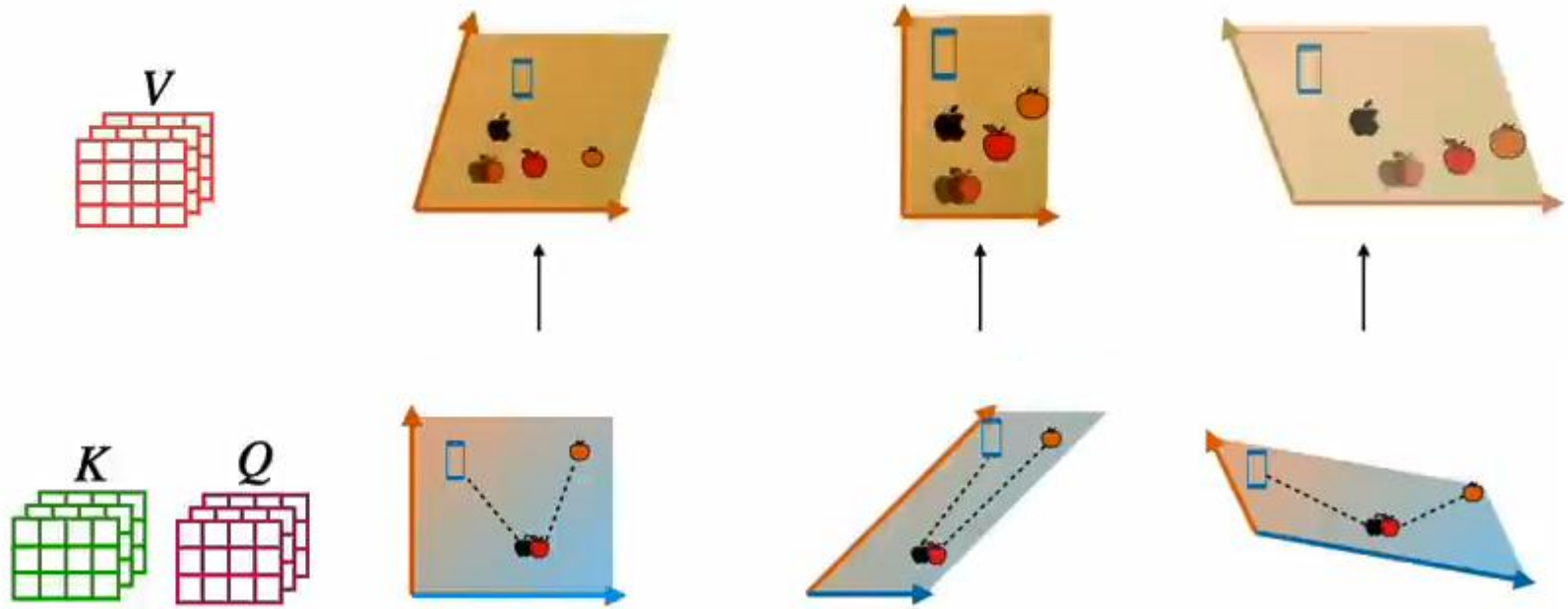


Multi-head attention

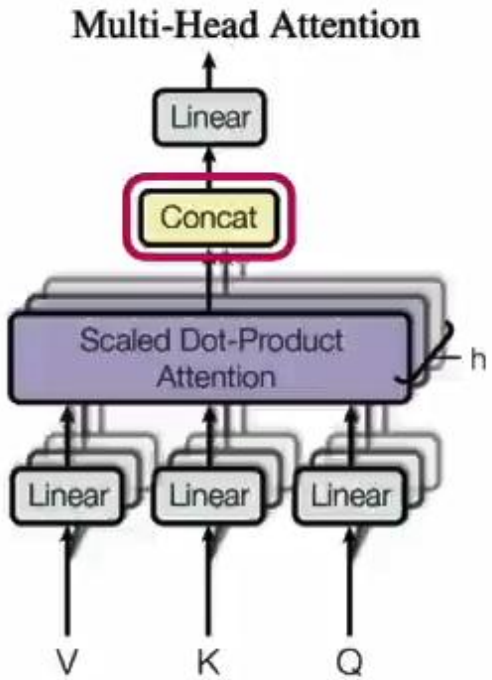


$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

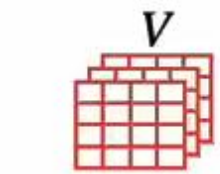
As we said before, this V matrix transform these embedding where we calculate the similarities into embeddings where we can move the words around

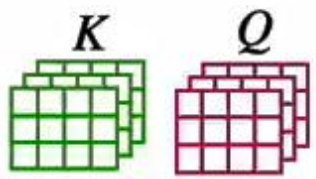


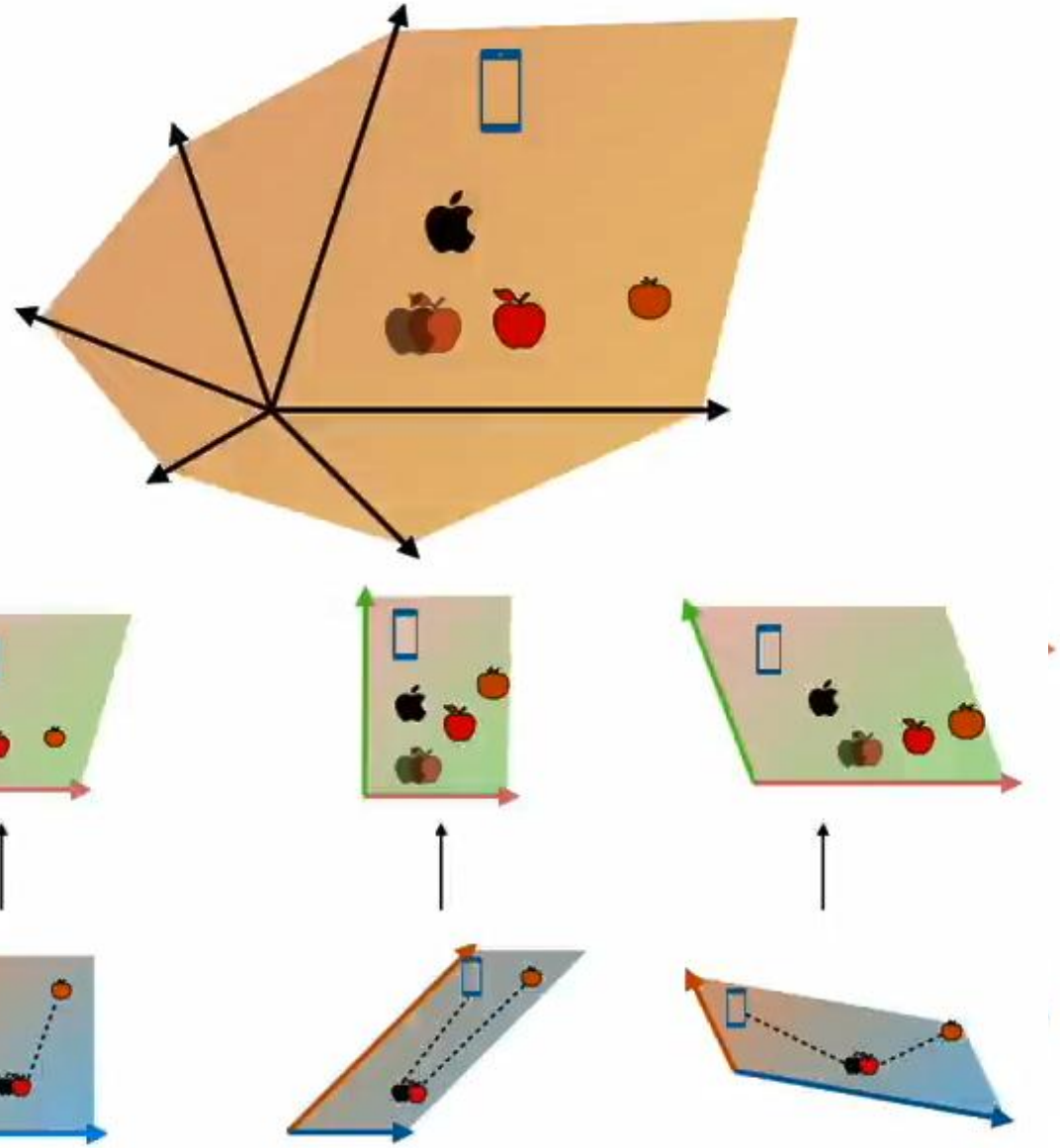
Multi-head attention



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

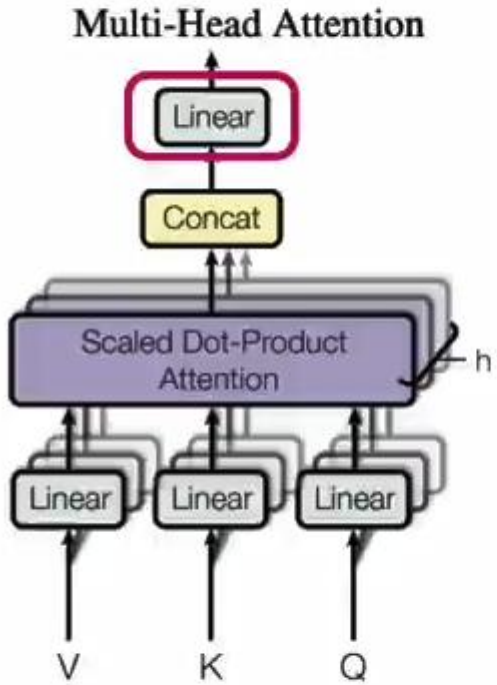




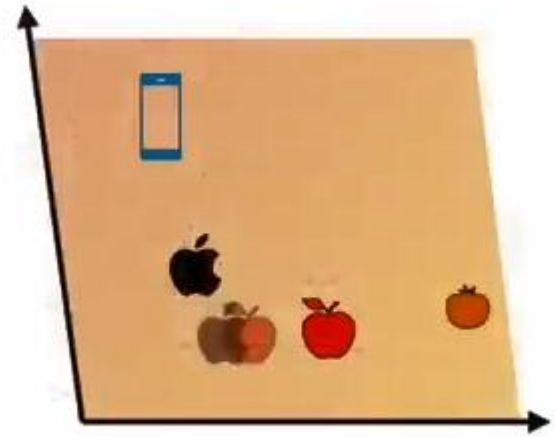


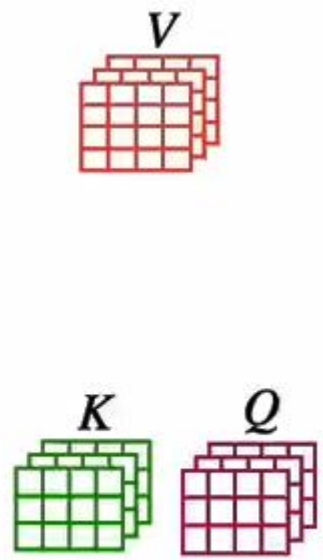
The concat the result, and I obtain an embeddings of 6 dimension. To reconduct this into lower dimension I will apply this Linear step (rectangular matrix)

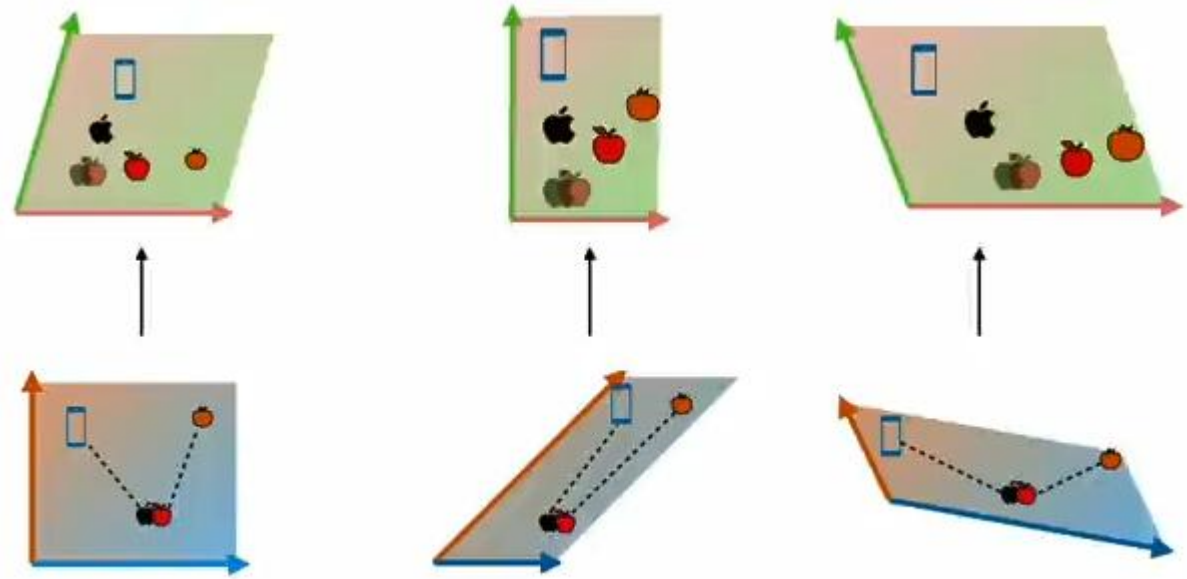
Multi-head attention



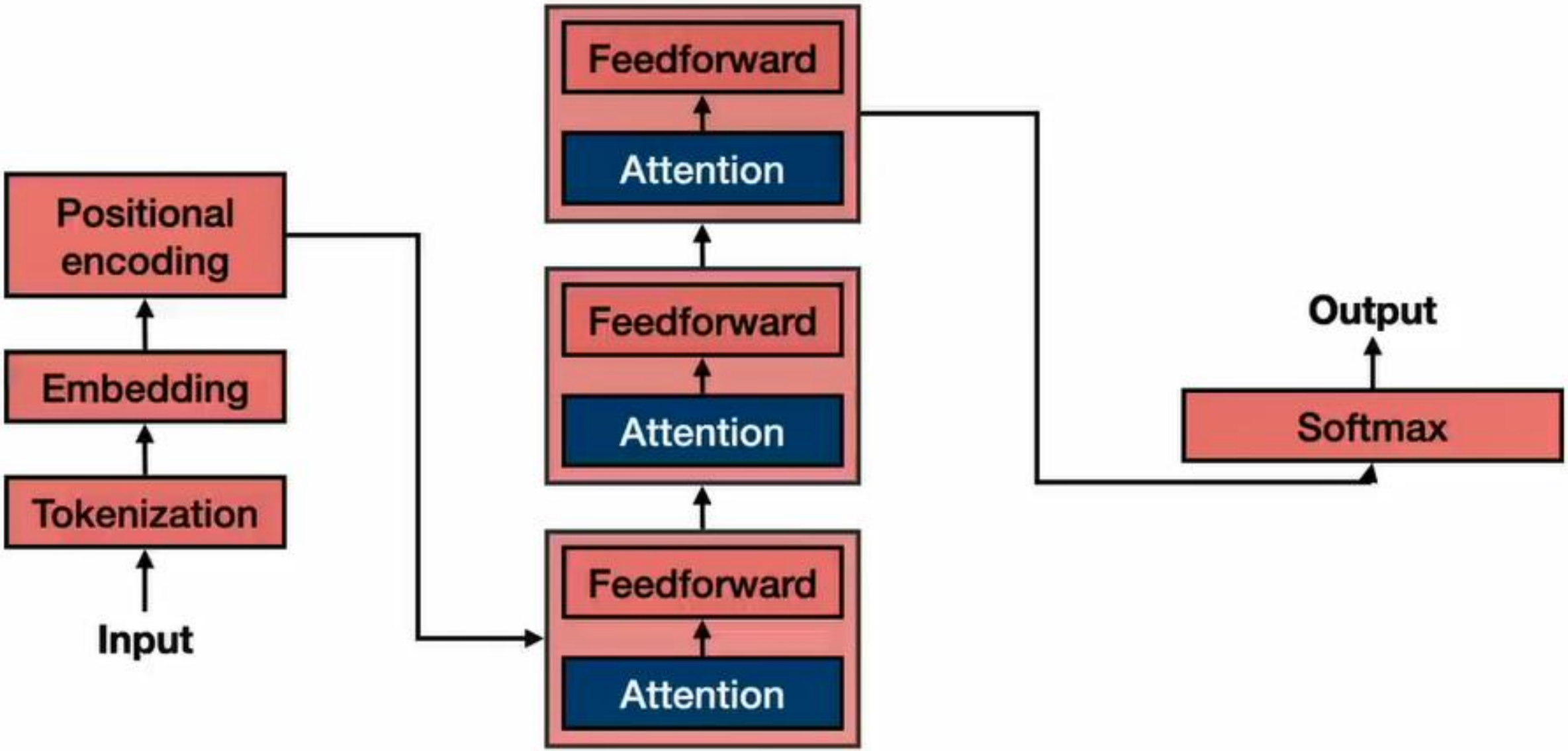
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$







Weights get trained with the transformer model




That's all Folks!