Transformers

Jose González-Abad
Instituto de Física de Cantabria (CSIC-UC)

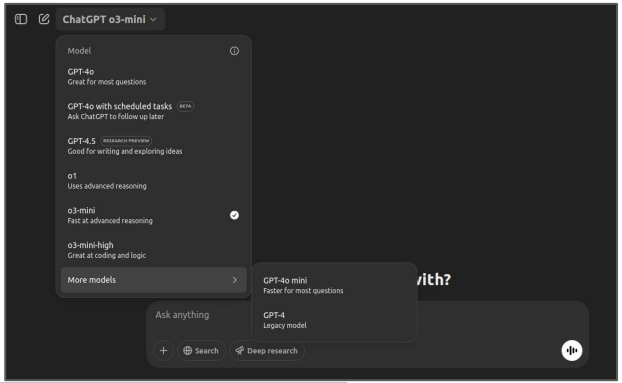
gonzabad@ifca.unican.es
jgonzalezab

Master Data Science (UIMP-UC)
March 2025

Origin of the Attention Mechanism

Most of the revolutionary advances in deep learning in recent years rely on **one simple yet powerful mechanism**

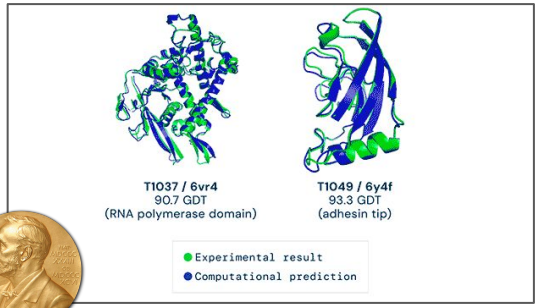
Large language models



Video generation

Image segmentation

Protein folding



Origin of the Attention Mechanism

It all started in **2017** at the **Neural Information Processing System (NIPS) conference** in Long Beach, California

Where a group of researchers from **Google** and **UoT** presented a **10-pages paper**

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Origin of the Attention Mechanism

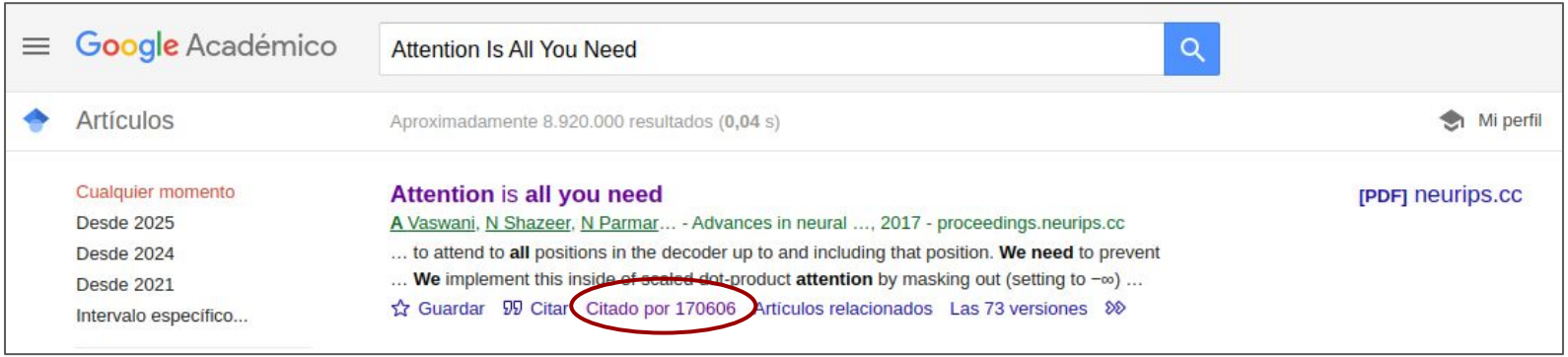
Typically, **research papers receive between 10 and 100 citations***

*This is illustrative as it highly depends on various factors such as the field of study

https://github.com/KentoNishi/awesome-all-you-need-papers

Origin of the Attention Mechanism

Typically, **research papers receive between 10 and 100 citations***; however, this particular paper has accumulated over **150,000 citations** to date

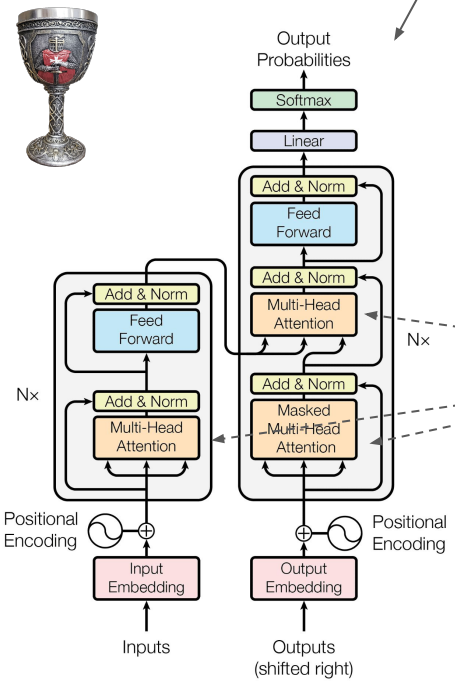


*This is illustrative as it highly depends on various factors such as the field of study

https://github.com/KentoNishi/awesome-all-you-need-papers

Origin of the Attention Mechanism

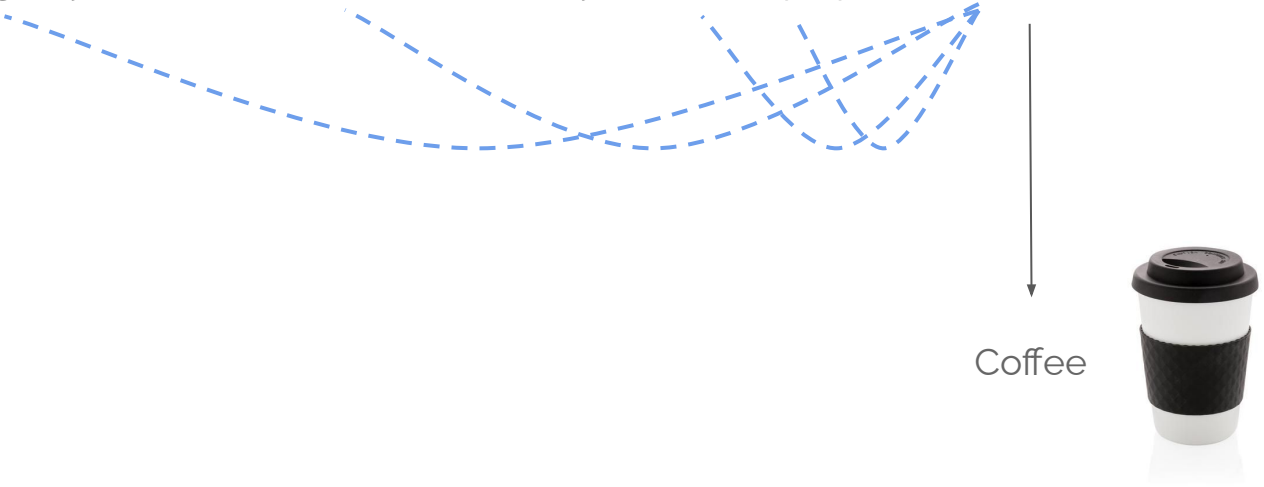
Its main contribution are **transformers**, which are based on the **attention mechanism**



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

What is attention?

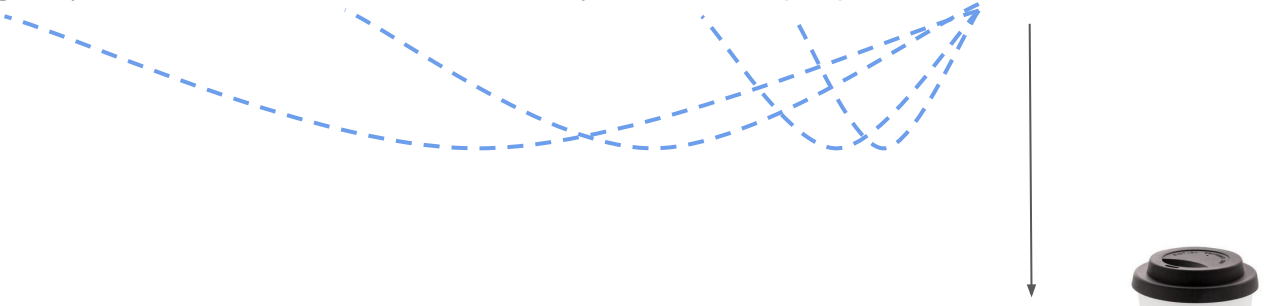
After a long day at work, she sat down to enjoy a warm cup of _____



The **attention mechanism** enables the deep learning model to determine **which parts of the context to focus on**

What is attention?

After a long day at work, she sat down to enjoy a warm cup of _____



Coffee

In the following, we will explain how the transformer is applied to learn this task

Tokenization

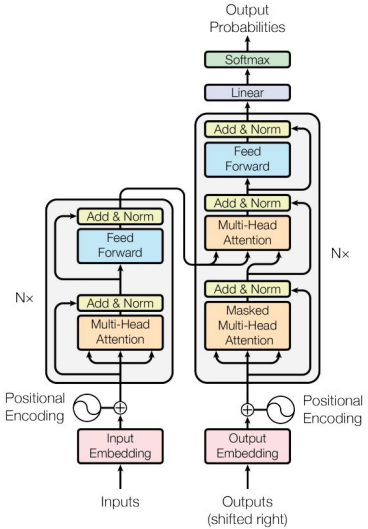


Figure 1: The Transformer - model architecture.

To **understand the transformer architecture**, we will examine each of its elements in detail as we complete the **following task**

After a long day at work, she sat down to enjoy a warm cup of _____

Tokenization

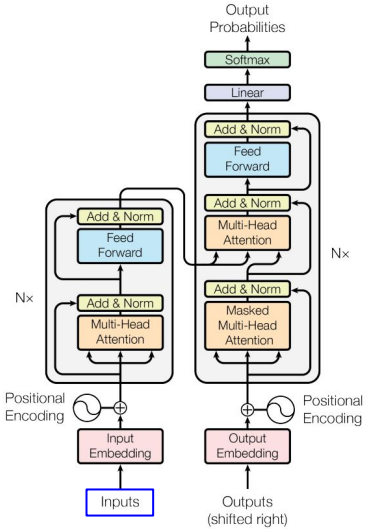
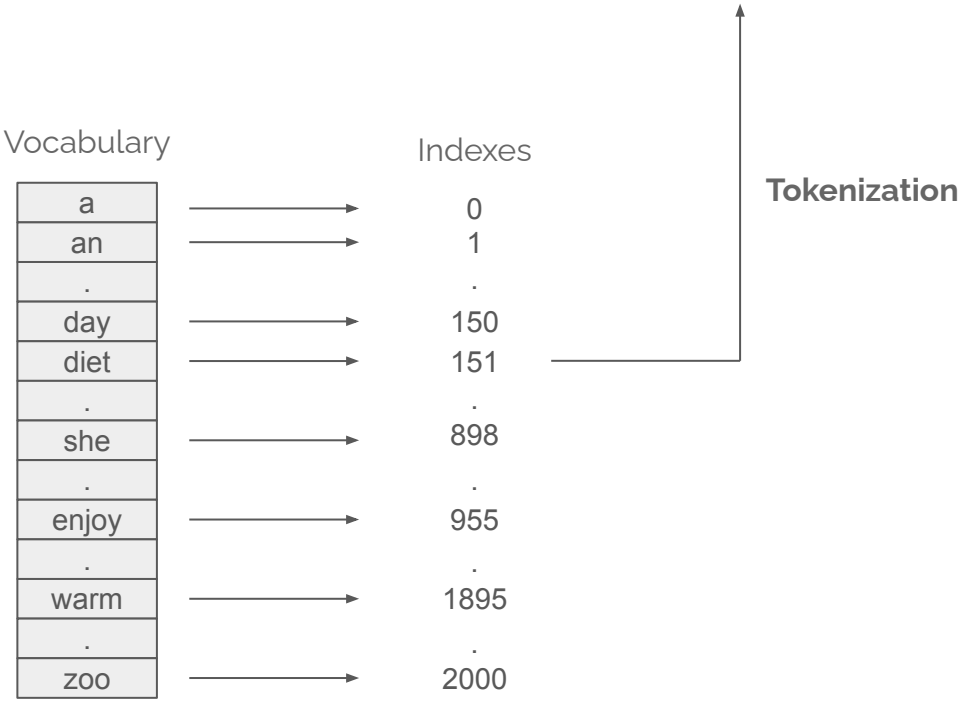


Figure 1: The Transformer - model architecture.

After a long day at work, she sat down to enjoy a warm cup of _____

22 0 432 150 15 1700 898 900 99 1200 501 0 1895 203 332



Input embedding

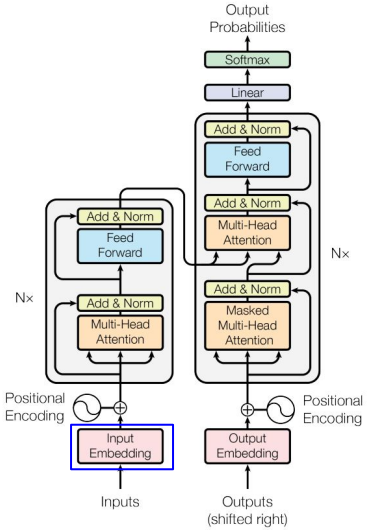
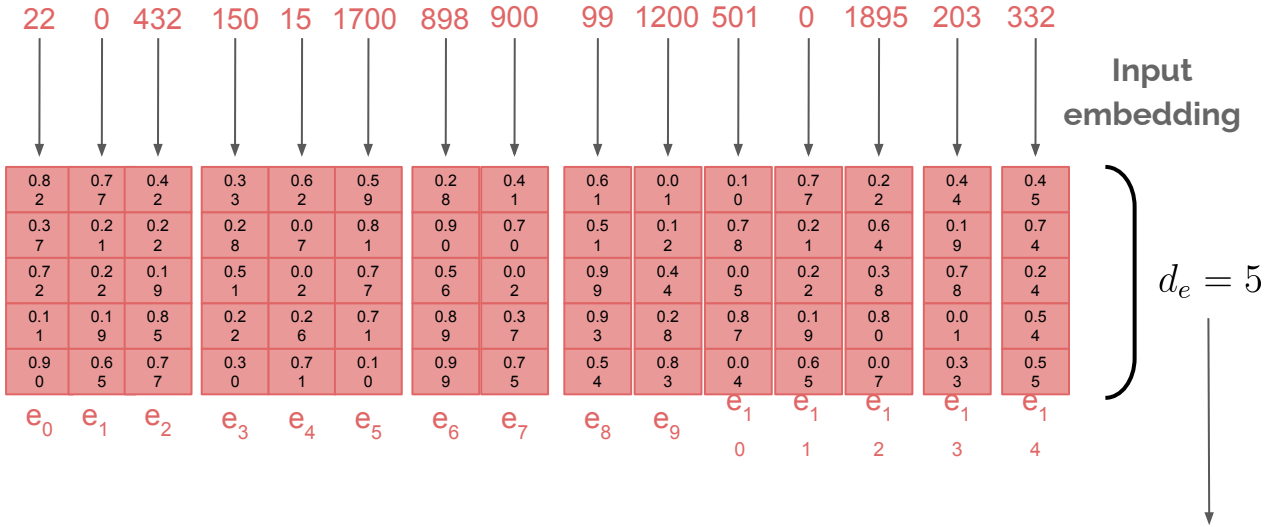


Figure 1: The Transformer - model architecture.

After a long day at work, she sat down to enjoy a warm cup of _____



This embeddings are generally **learned** through a matrix $\mathbf{W}_e$

The embedding dimension tends to be higher

Input embedding

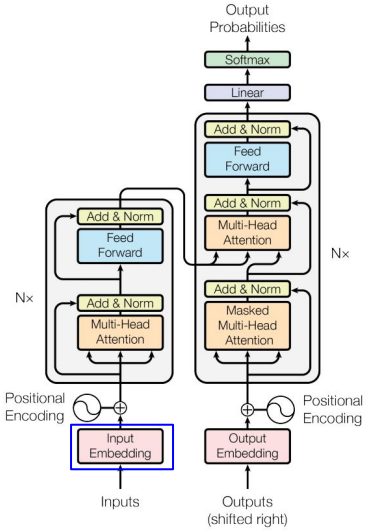
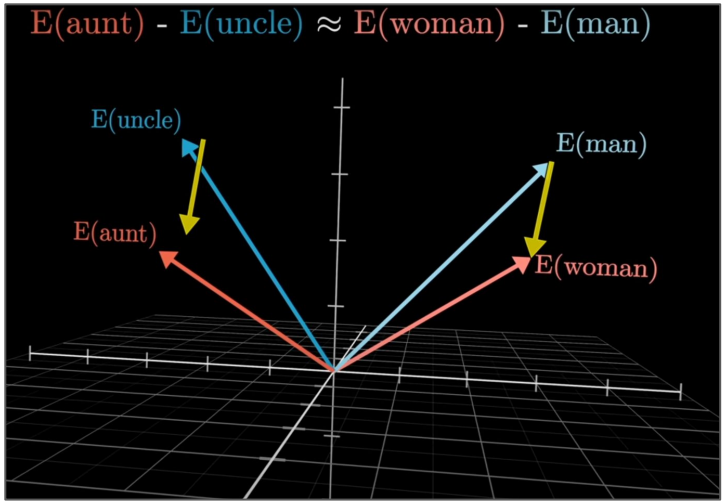
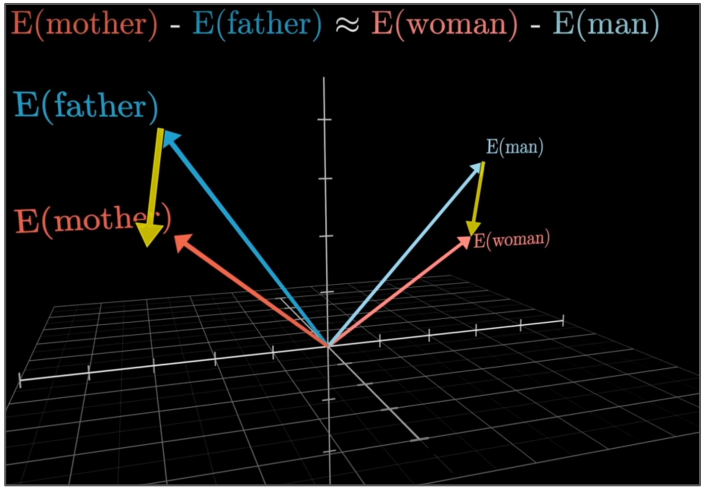


Figure 1: The Transformer - model architecture.

What do these embeddings **represent**?





Input embedding

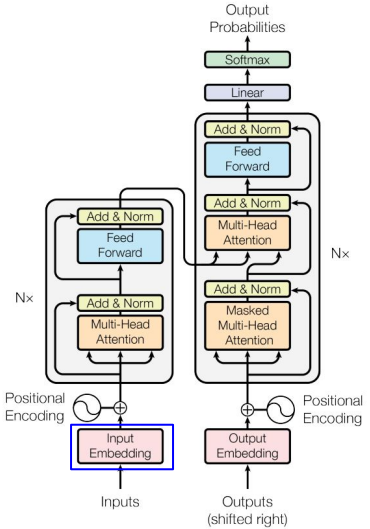
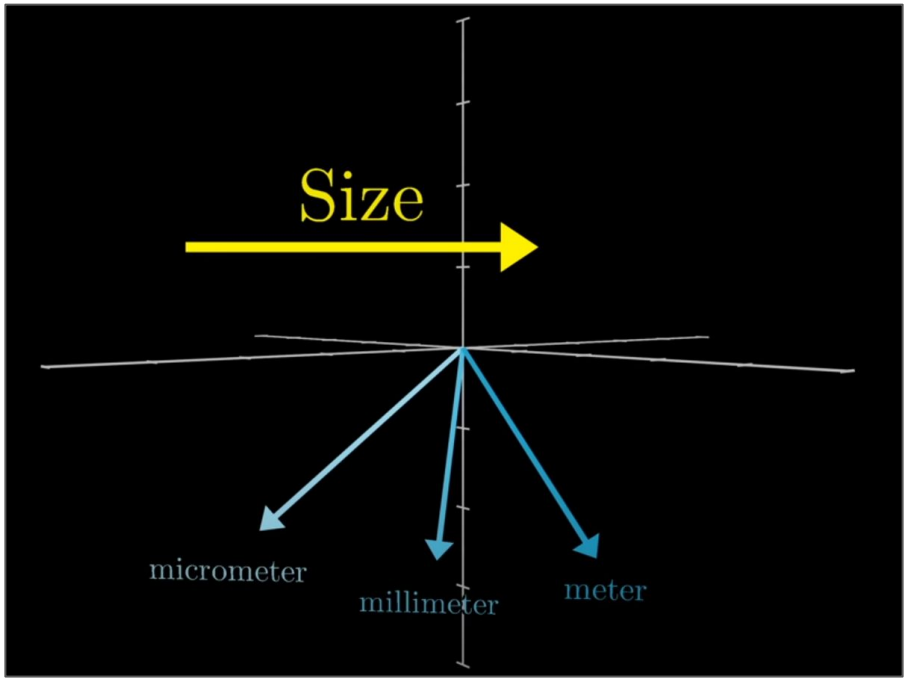


Figure 1: The Transformer - model architecture.

What do these embeddings **represent**?



Input embedding

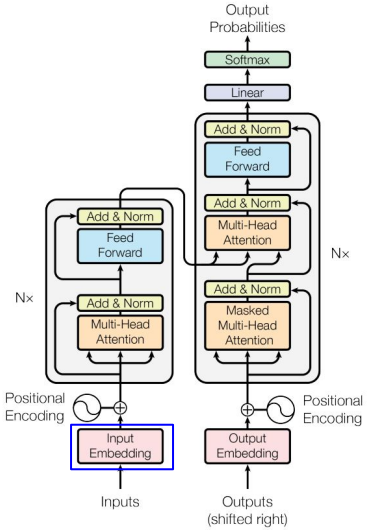


Figure 1: The Transformer - model architecture.

After applying the **attention mechanism**, we expect the embeddings to capture a more **complex representation of the context**

*After a long day at work, she sat down to enjoy a **warm cup** of _____*

Warm cup

Cup

We **transform** the meaning of the **embedding** based on the **context**

Positional encoding

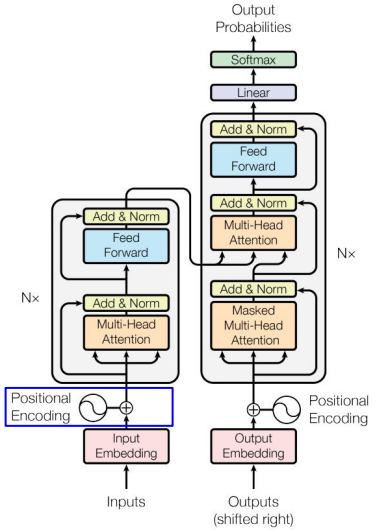
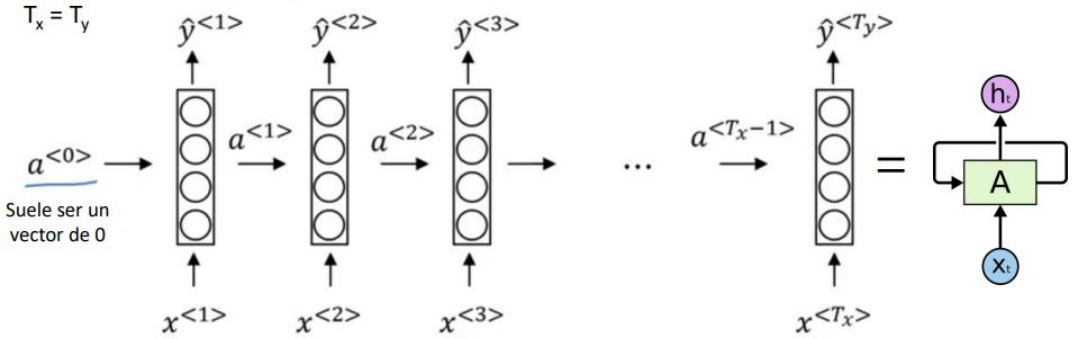


Figure 1: The Transformer - model architecture.

Other models that process sequences can take the order of inputs into account



However, transformers consider **all inputs at once**

How can we **enforce an order on the inputs** for transformers?

Positional encoding

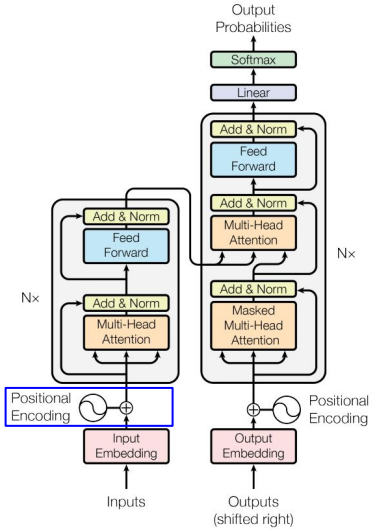
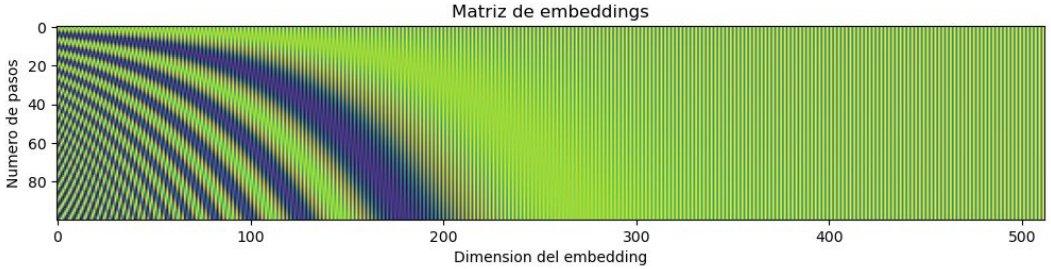


Figure 1: The Transformer - model architecture.

After a long day at work, she sat down to enjoy a warm cup of _____

Input embedding

0.82	0.77	0.42	0.33	0.62	0.59	0.28	0.41	0.61	0.01	0.10	0.77	0.22	0.44	0.45
0.37	0.21	0.22	0.28	0.07	0.81	0.90	0.70	0.51	0.12	0.78	0.21	0.64	0.19	0.74
0.72	0.22	0.19	0.51	0.02	0.77	0.56	0.02	0.99	0.44	0.05	0.22	0.38	0.78	0.24
0.11	0.19	0.85	0.22	0.26	0.71	0.89	0.37	0.93	0.28	0.87	0.19	0.80	0.01	0.54
0.90	0.65	0.77	0.30	0.71	0.10	0.99	0.75	0.54	0.83	0.04	0.65	0.07	0.33	0.55
e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}



Positional encoding

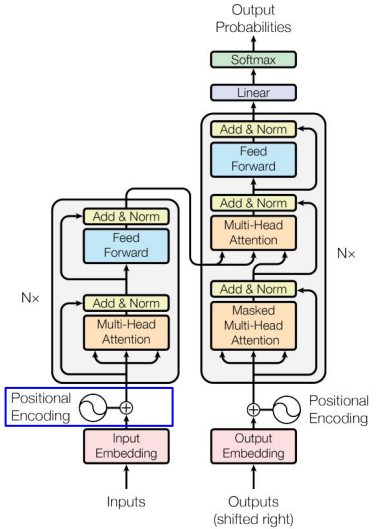
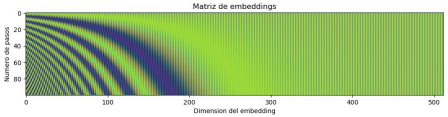


Figure 1: The Transformer - model architecture.

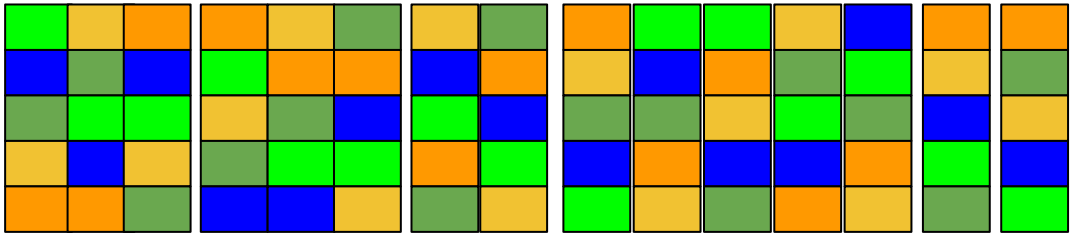


After a long day at work, she sat down to enjoy a warm cup of _____

Input embedding

0.82	0.77	0.42	0.33	0.62	0.59	0.28	0.41	0.61	0.01	0.10	0.77	0.22	0.44	0.45
0.37	0.21	0.22	0.28	0.07	0.81	0.90	0.70	0.51	0.12	0.78	0.21	0.64	0.19	0.74
0.72	0.22	0.19	0.51	0.02	0.77	0.56	0.02	0.99	0.44	0.05	0.22	0.38	0.78	0.24
0.11	0.19	0.85	0.22	0.26	0.71	0.89	0.37	0.93	0.28	0.87	0.19	0.80	0.01	0.54
0.90	0.65	0.77	0.30	0.71	0.10	0.99	0.75	0.54	0.83	0.04	0.65	0.07	0.33	0.55
e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}

+



Word embeddings **aware of its position** within the sequence

Attention mechanism

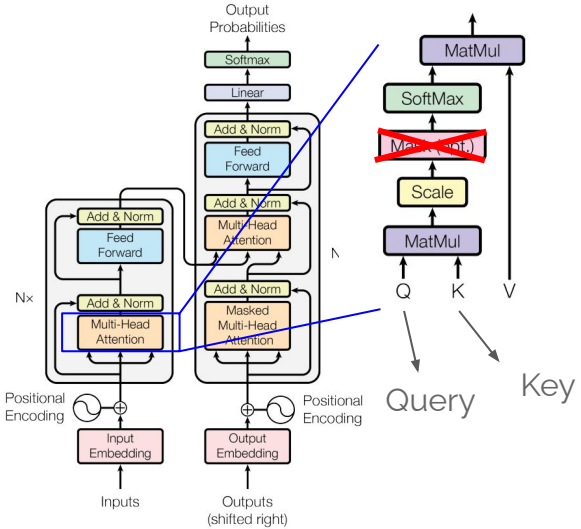
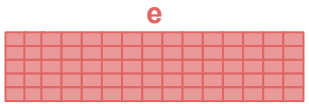


Figure 1: The Transformer - model architecture.



Attention mechanism

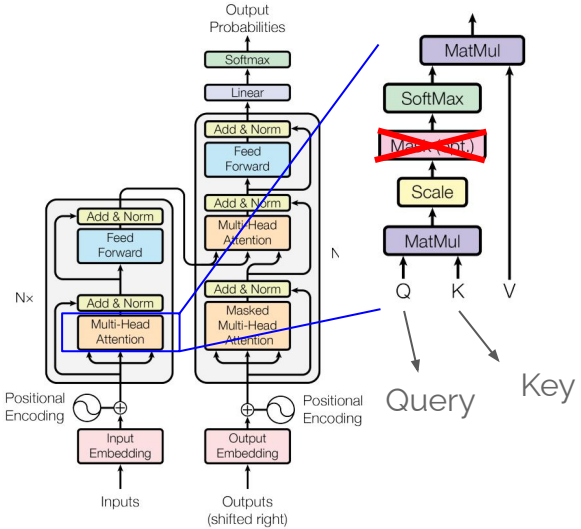
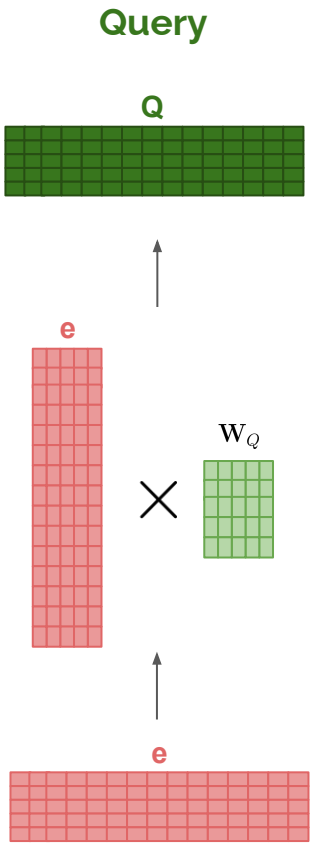
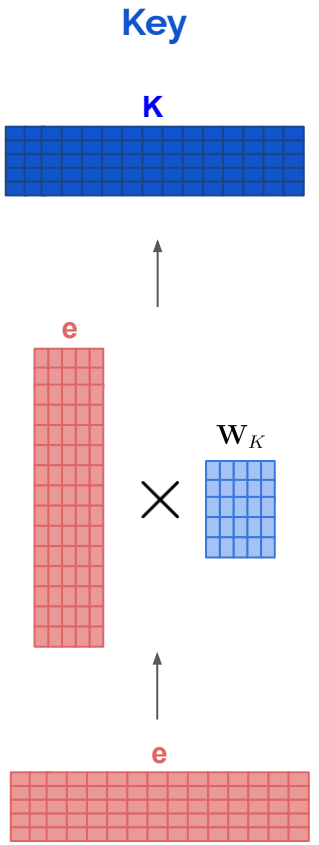


Figure 1: The Transformer - model architecture.





Attention mechanism

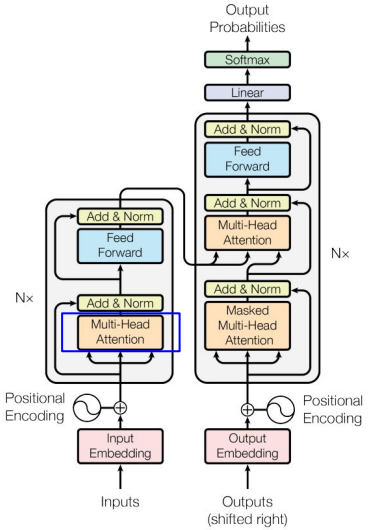
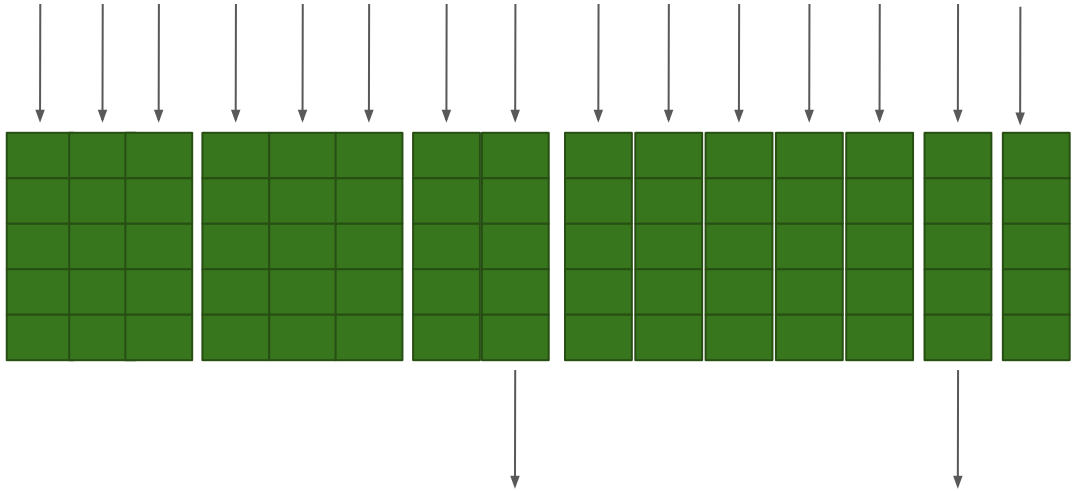


Figure 1: The Transformer - model architecture.

What do **queries** and **keys** **represent**?

After a long day at work, she sat down to enjoy a warm cup of _____



Query

Any information about **who** performed the action?

Any **adjectives** in front of me?

Attention mechanism

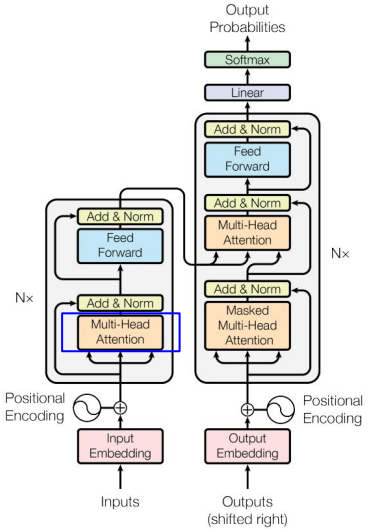
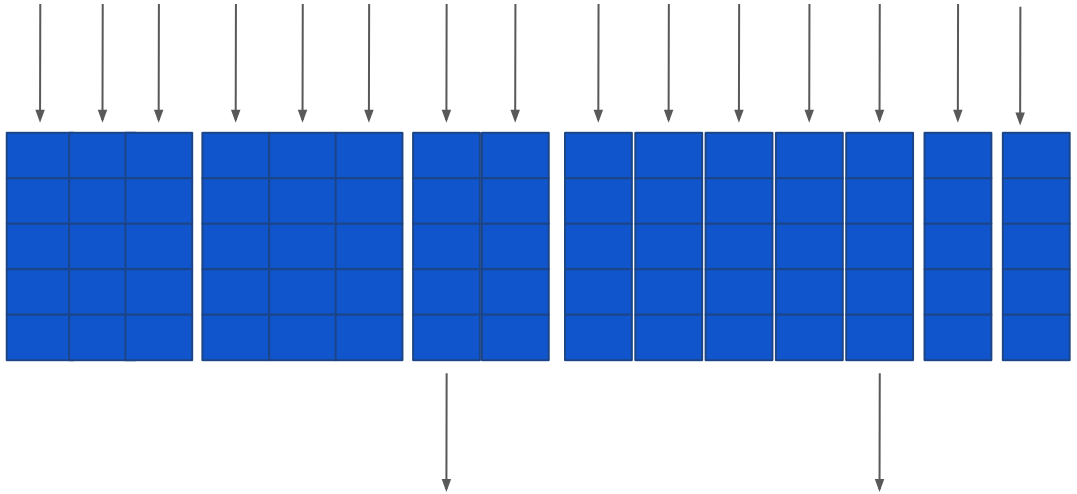


Figure 1: The Transformer - model architecture.

What do **queries** and **keys** **represent**?

After a long day at work, she sat down to enjoy a warm cup of _____



Key

I refer to people!

I am an adjective!

Attention mechanism

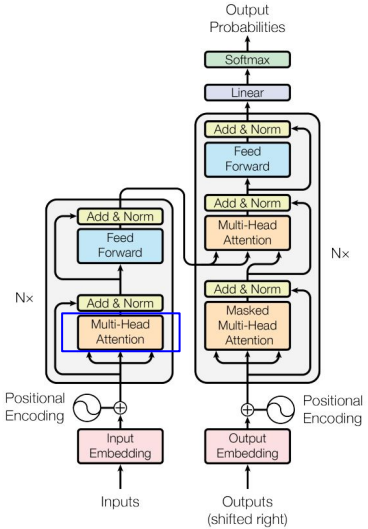
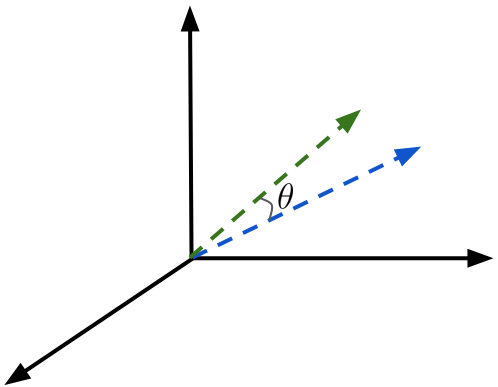


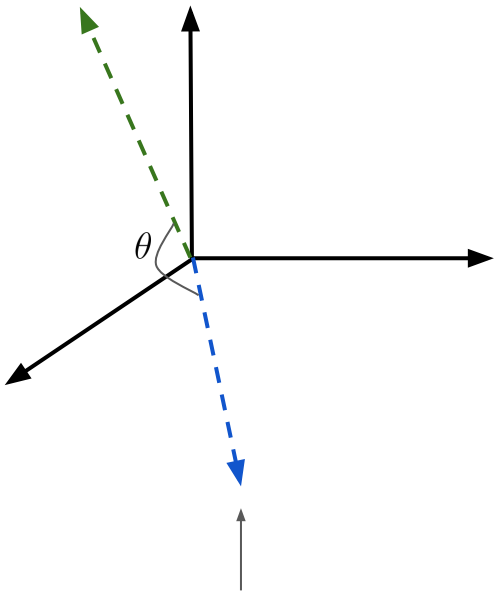
Figure 1: The Transformer - model architecture.

How can we compute similarity between **query** and **key** vectors?

High similarity



Low similarity



Attention mechanism

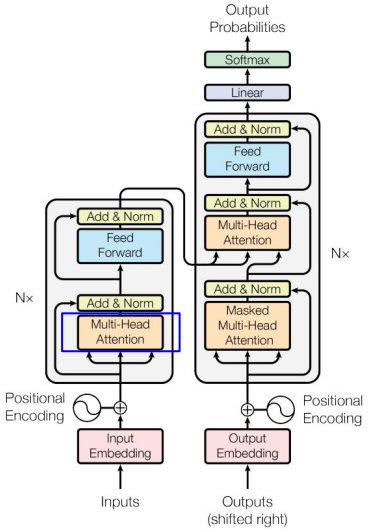
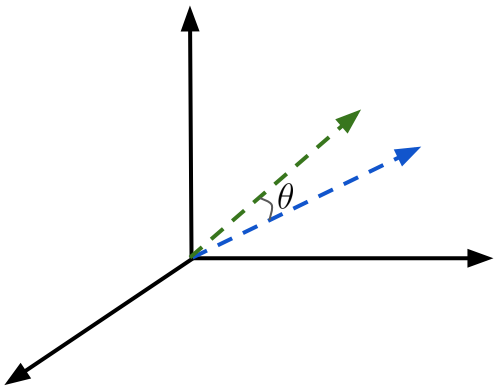


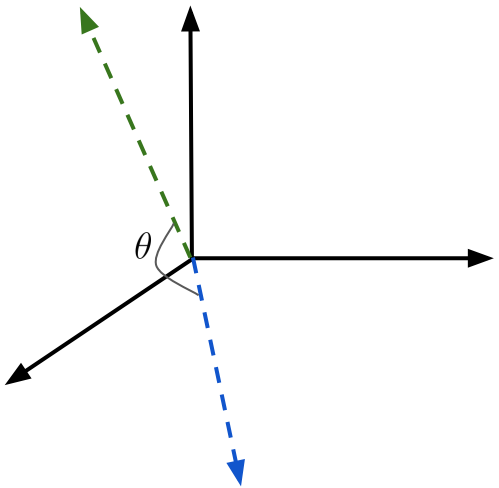
Figure 1: The Transformer - model architecture.

How can we compute similarity between **query** and **key** vectors?

High similarity



Low similarity



Cosine similarity

$$\mathrm{Cos}(A, B) = \frac{A \cdot B^T}{|A| \cdot |B|}$$

High values

Low values

Attention mechanism

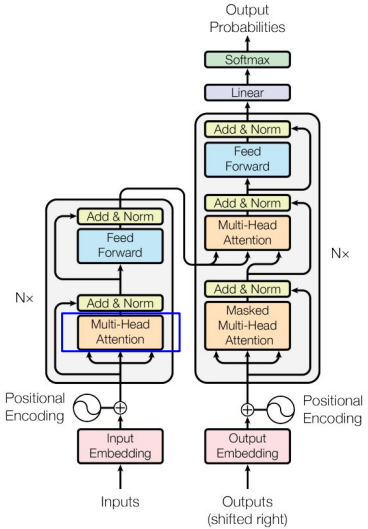


Figure 1: The Transformer - model architecture.

How can we compute similarity between **query** and **key** vectors?

$$\mathrm{Cos}(A,B) = \frac{A \cdot B^T}{|A| \cdot |B|} \longrightarrow \mathrm{Similarity}(Q,K) = \frac{QK^T}{\sqrt{d_K}}$$

Dimension of the key

Attention mechanism

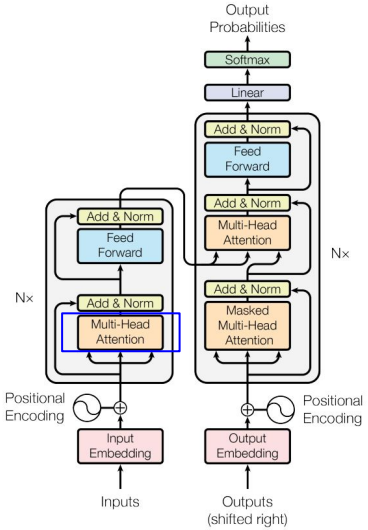
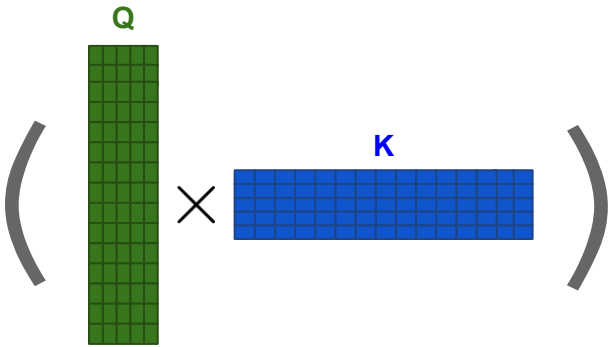


Figure 1: The Transformer - model architecture.

How can we compute similarity between **query** and **key** vectors?

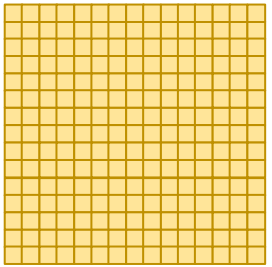
$$\mathrm{Cos}(A,B) = \frac{A \cdot B^T}{|A| \cdot |B|} \longrightarrow \mathrm{Similarity}(Q,K) = \frac{QK^T}{\sqrt{d_K}}$$

Dimension of the key



$$\mathrm{Similarity}(Q,K) = \frac{\left(\quad \times \quad\right)}{\sqrt{d_K}} =$$

Attention matrix



Attention mechanism

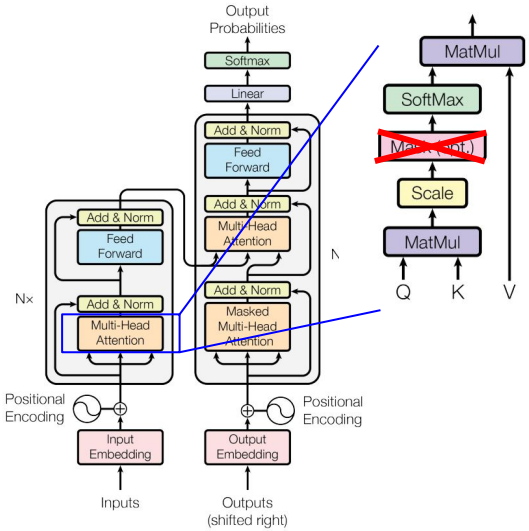
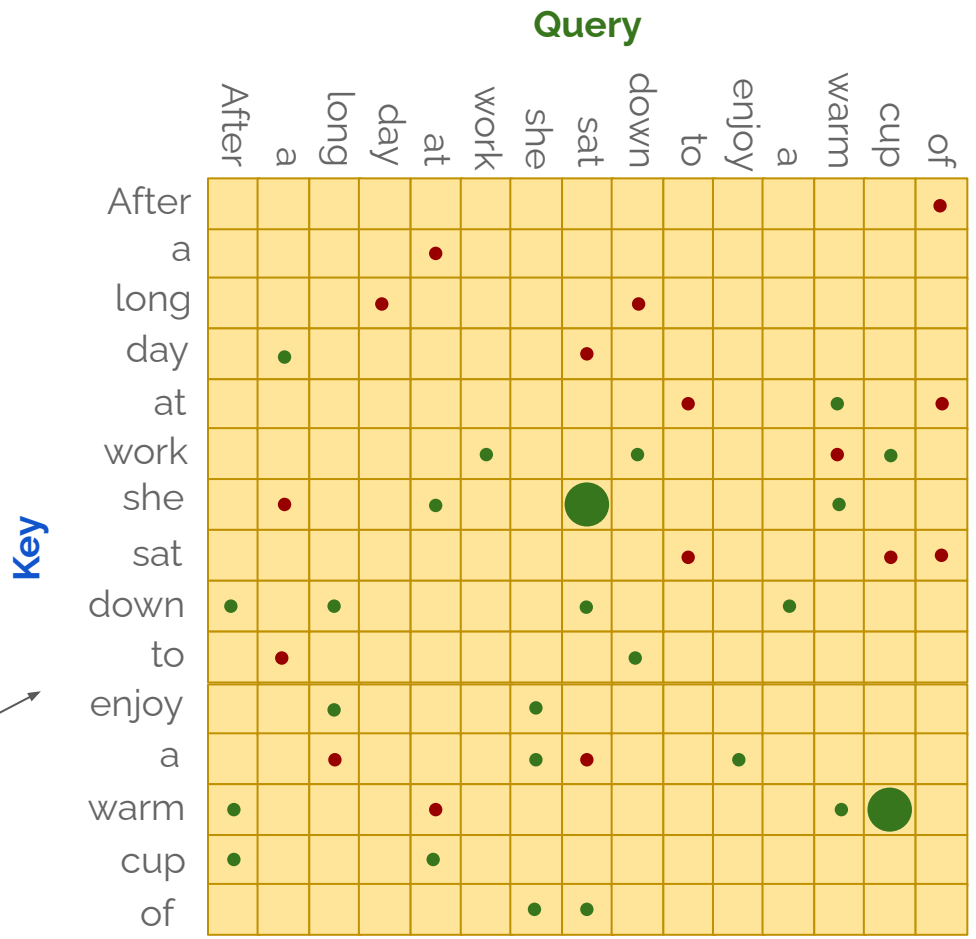


Figure 1: The Transformer - model architecture.

$$\frac{QK^T}{\sqrt{d_K}}$$



Attention mechanism

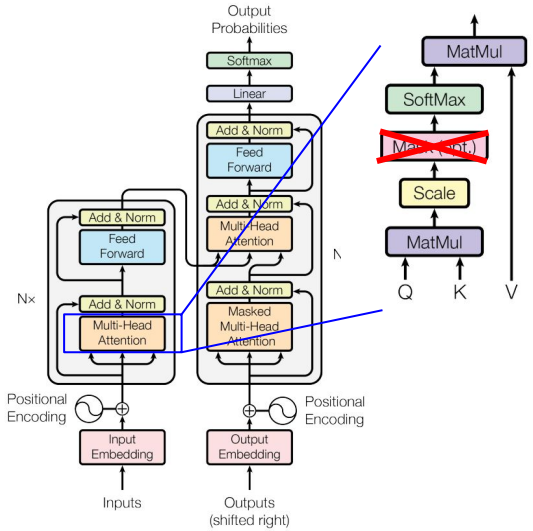


Figure 1: The Transformer - model architecture.

$$\frac{QK^T}{\sqrt{d_K}}$$

Key

Query

Key	After	a	long	day	at	work	she	sat	down	to	enjoy	a	warm	cup	of
After								0.01							
a								0.1							
long								-0.2							
day								-0.01							
at								-0.71							
work								0.1							
she								3.07							
sat								0.89							
down								-0.09							
to								0.15							
enjoy								-0.5							
a								-0.12							
warm								0.2							
cup								0.29							
of								0.61							

Attention mechanism

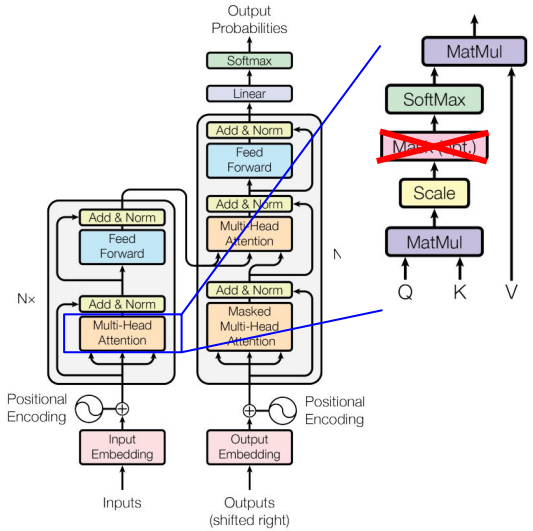
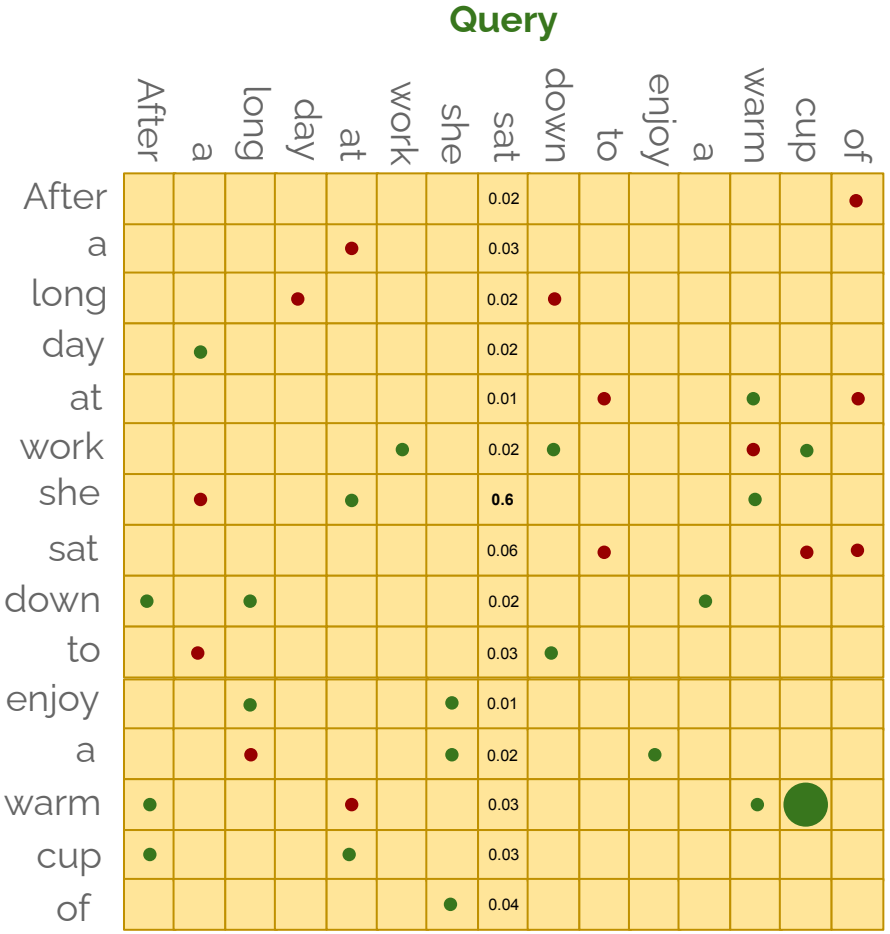


Figure 1: The Transformer - model architecture.

$$\text{softmax}(\frac{QK^T}{\sqrt{d_K}})$$

Key

Query



Attention mechanism

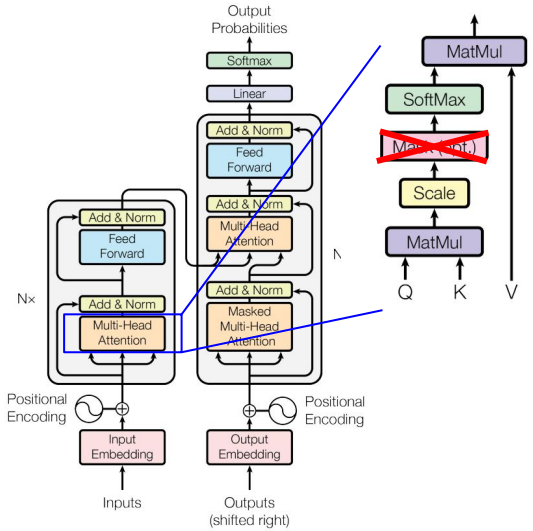
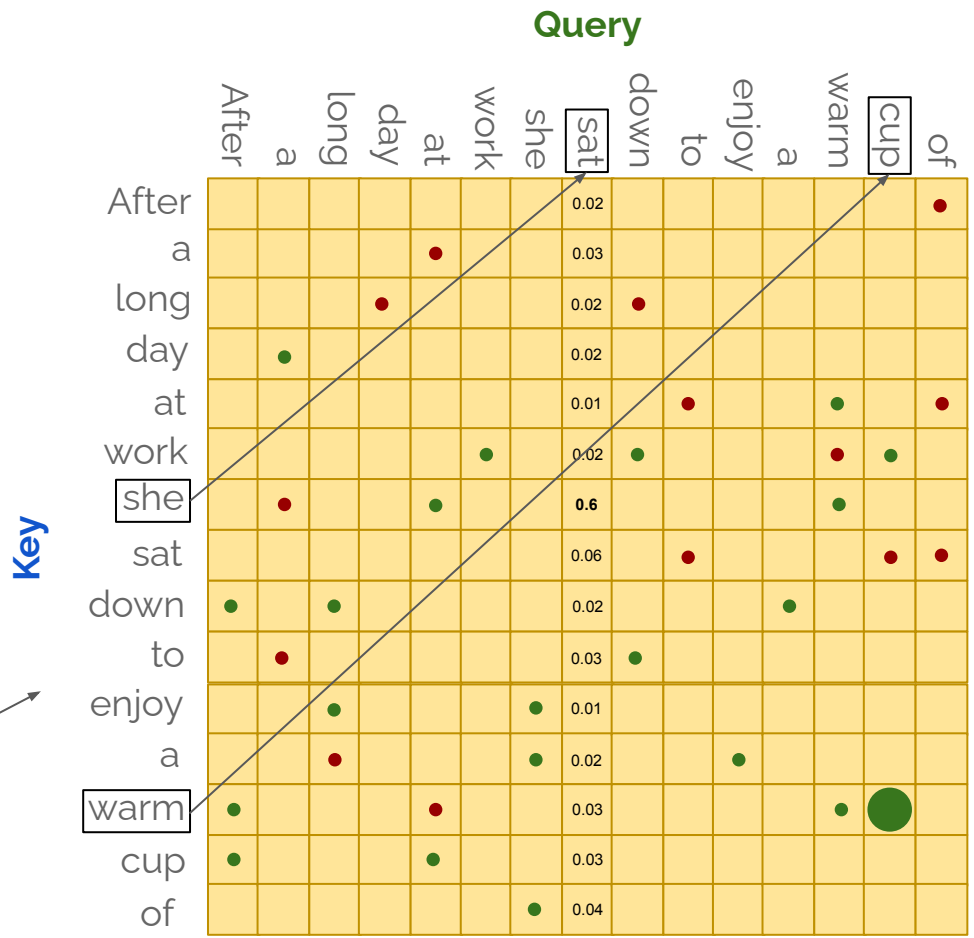


Figure 1: The Transformer - model architecture.

$$\text{softmax}(\frac{QK^T}{\sqrt{d_K}})$$



Attention mechanism

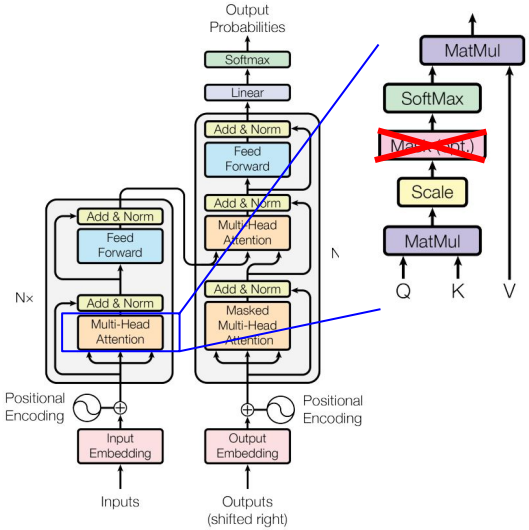


Figure 1: The Transformer - model architecture.

If we look more closely, we can see that we have almost **derived the attention formula**

Attention matrix → $\text{softmax}(\frac{QK^T}{\sqrt{d_K}})$

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

But, what about the **V**?

Attention mechanism

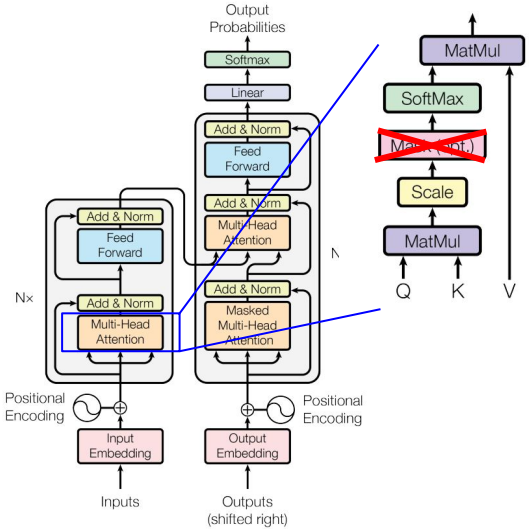


Figure 1: The Transformer - model architecture.

Similarly to how we generate the **query** and **key** vectors, we also generate the **value** vectors

Attention mechanism

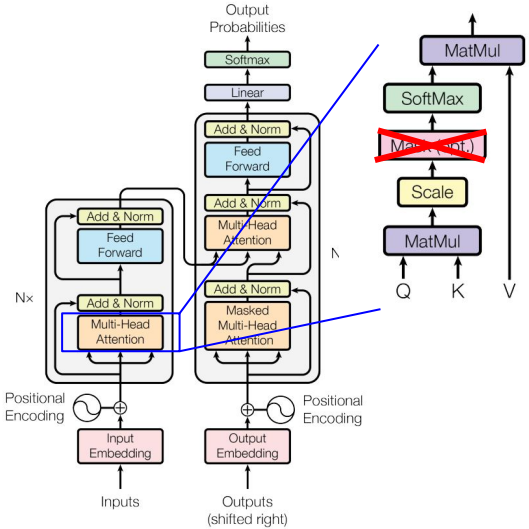
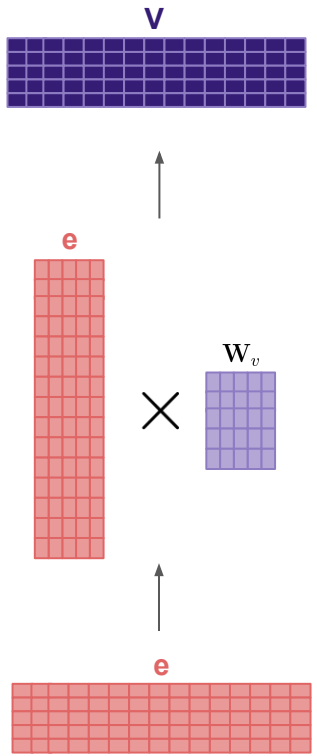


Figure 1: The Transformer - model architecture.

Similarly to how we generate the **query** and **key** vectors, we also generate the **value** vectors



A more refined and **complex** version of **e**

Attention mechanism

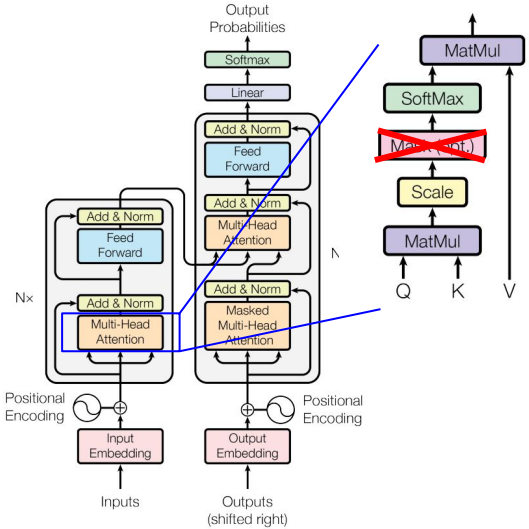
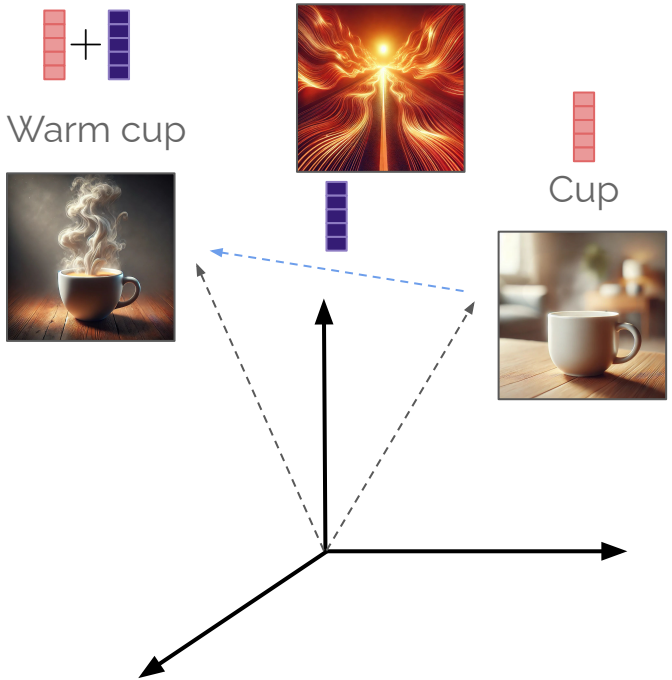


Figure 1: The Transformer - model architecture.

The **value** represents the step that the embeddings need to take to **update their meaning based on the context**



After a long day at work, she sat down to enjoy a warm cup of _____

Attention mechanism

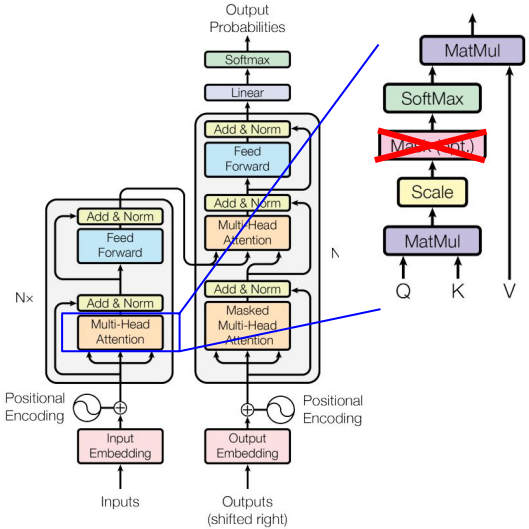
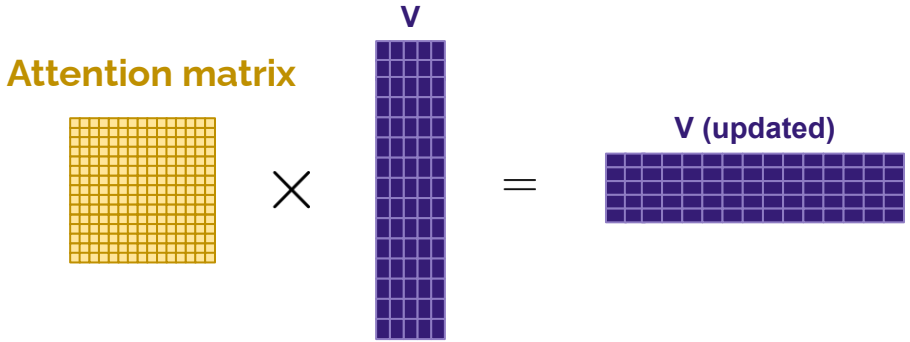


Figure 1: The Transformer - model architecture.

Before adding the **values** to the embedding we **multiply** them by the **attention matrix**



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

Attention mechanism

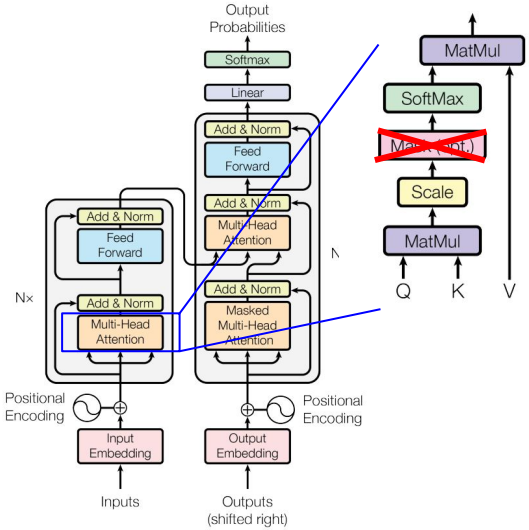
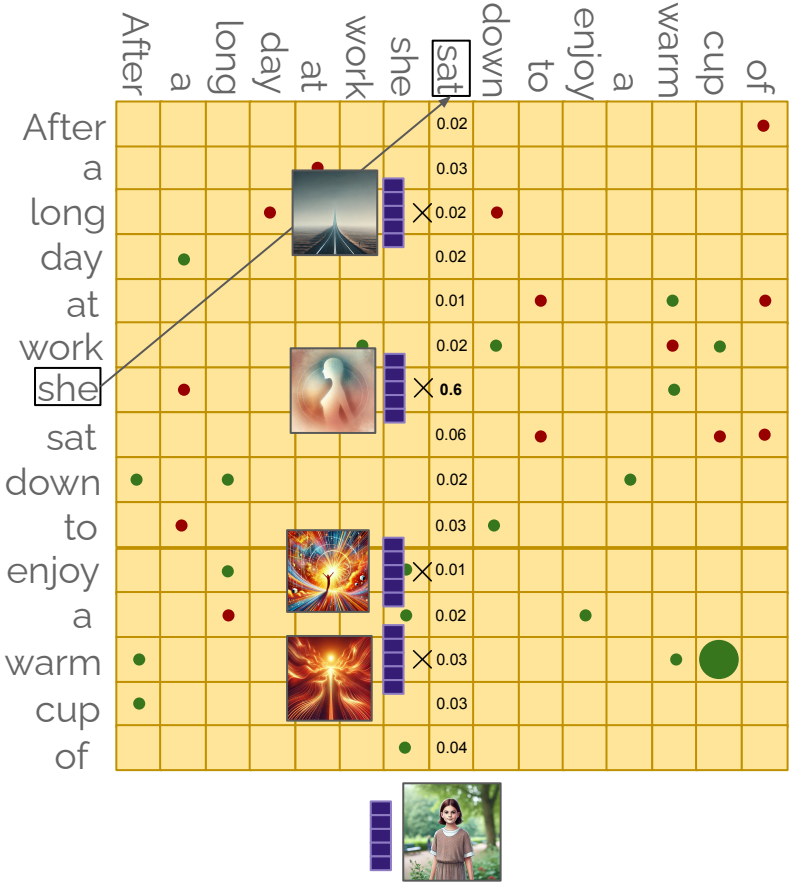


Figure 1: The Transformer - model architecture.



Attention mechanism

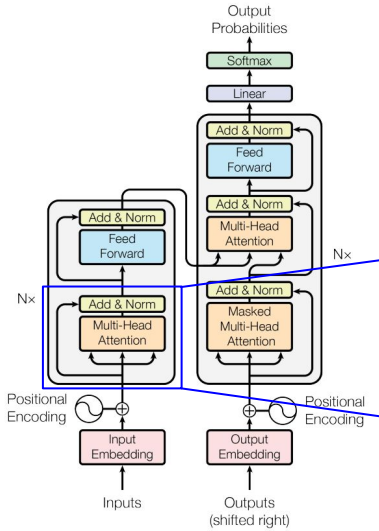
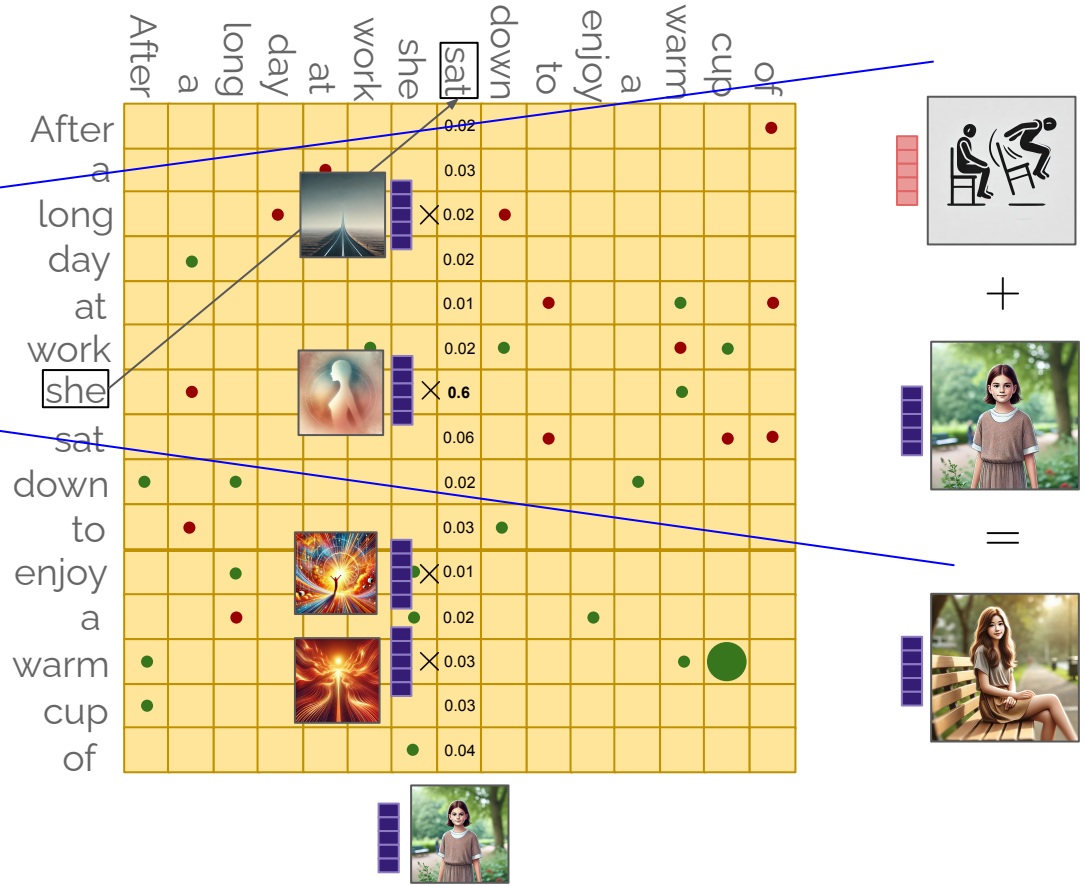


Figure 1: The Transformer - model architecture.



Attention mechanism

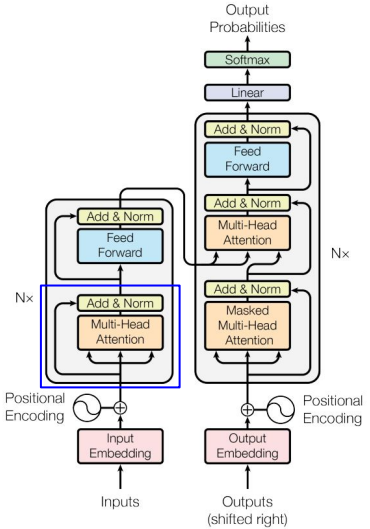
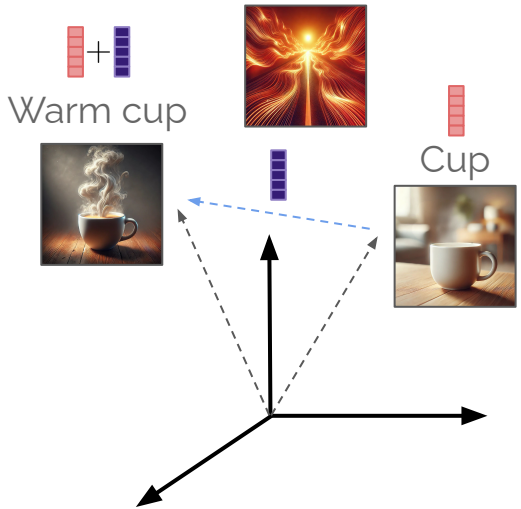


Figure 1: The Transformer - model architecture.

With the attention mechanism we **transform** the meaning of the **embedding** based on the **context**

*After a long day at work, she sat down to enjoy a **warm cup** of _____*



Attention mechanism

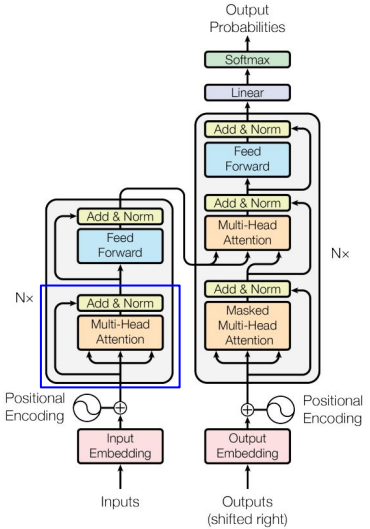
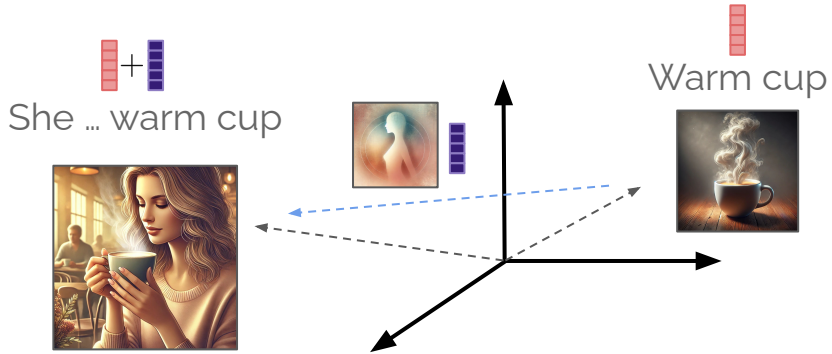


Figure 1: The Transformer - model architecture.

We can apply the attention mechanism to the **output of previous layers**, thereby building **more complex representations** of relationships between concepts

*After a long day at work, **she** sat down to enjoy a **warm cup** of _____*



Multi-Head Attention

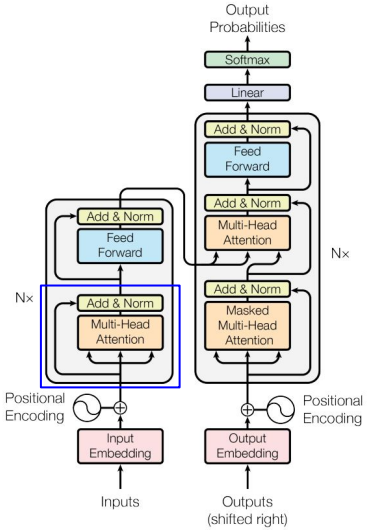
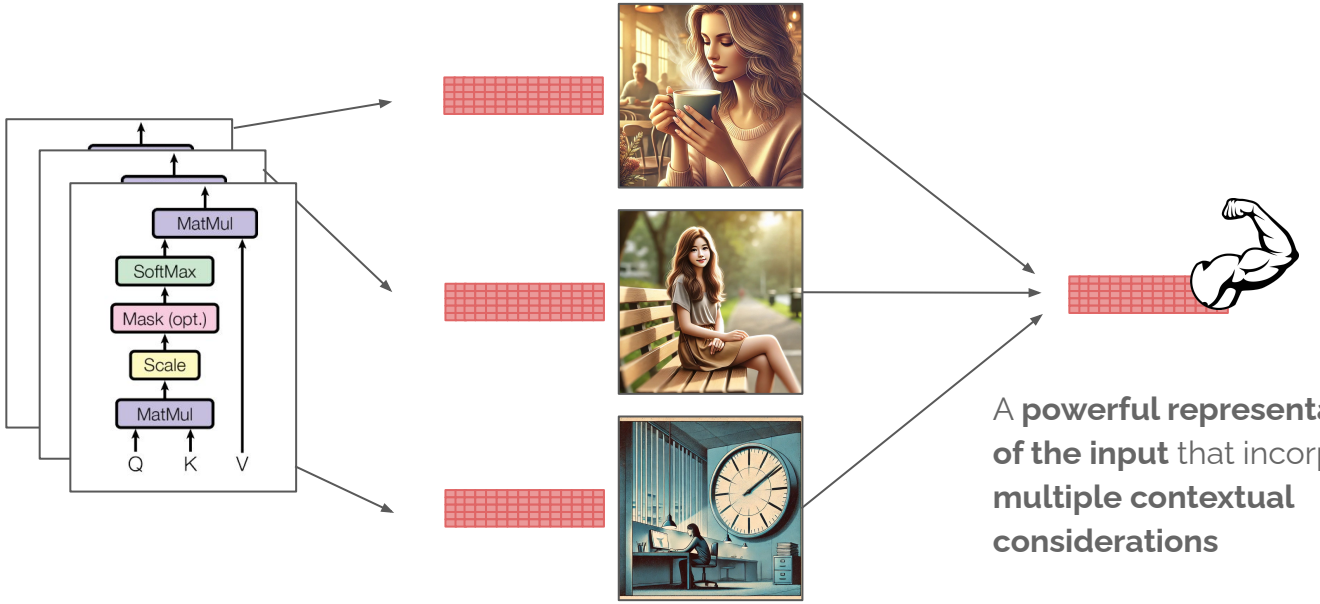


Figure 1: The Transformer - model architecture.

In transformers, multiple **attention mechanisms** are computed in **parallel**, a process known as **multi-head attention**



A **powerful representation of the input** that incorporates **multiple contextual considerations**

Encoder

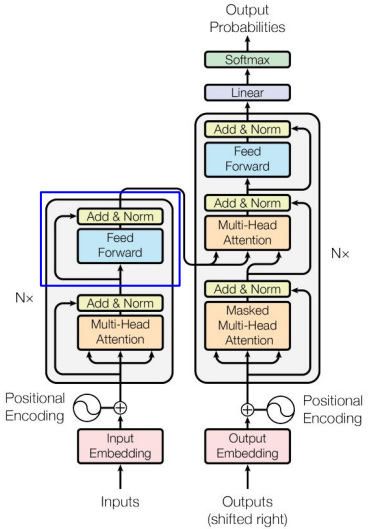
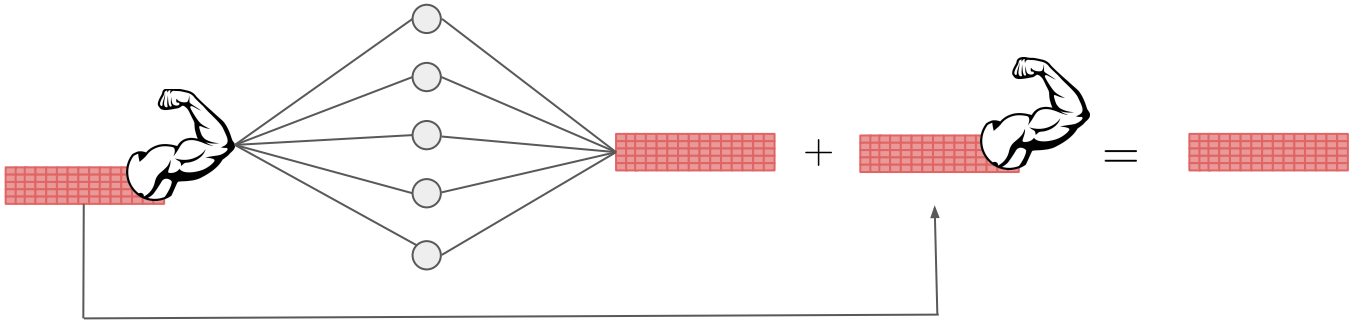


Figure 1: The Transformer - model architecture.

Finally, this representation is passed through a **fully-connected layer** through a **residual connection**



Encoder

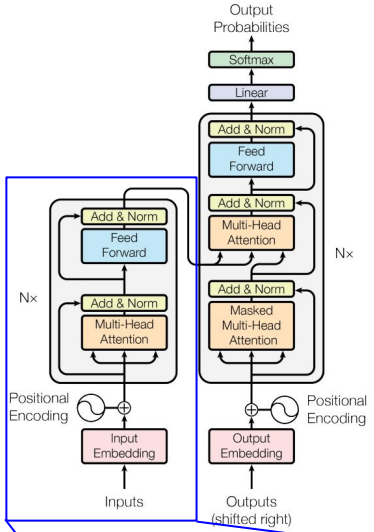
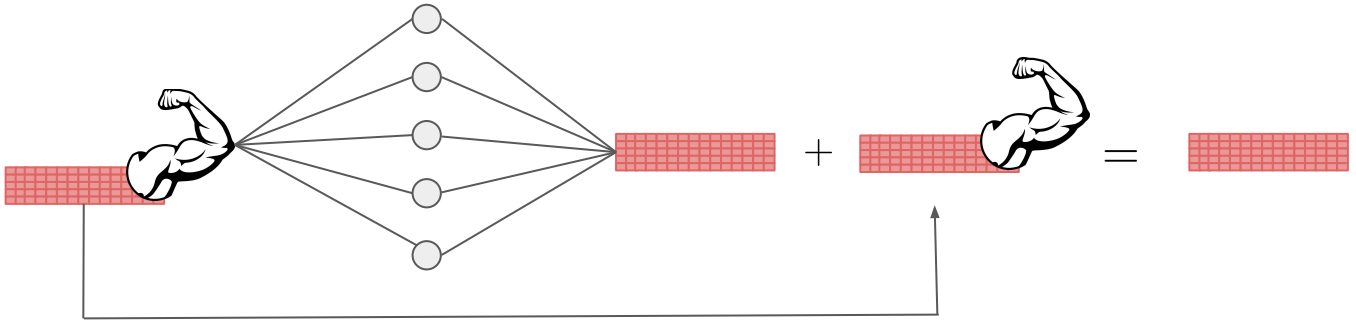


Figure 1: The Transformer - model architecture.

Finally, this representation is passed through a **fully-connected layer** through a **residual connection**



Encoder

This part of the transformer **encodes** the input while considering the context in multiple ways

Decoder

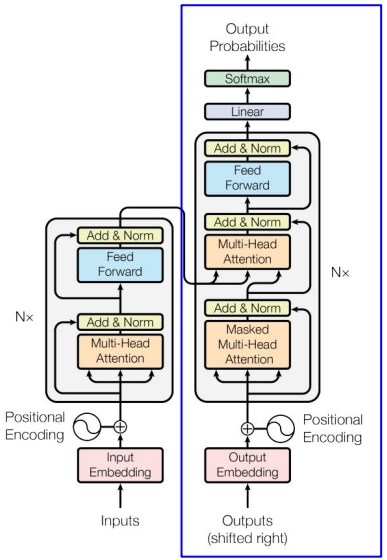


Figure 1: The Transformer - model architecture.

The other part of the transformer is generally referred to as the **decoder**. Its purpose is to **generate the next word in the sequence**

After a long day at work, she sat down to enjoy a warm cup of _____

To achieve this, it relies on the **embeddings processed and updated by the encoder**

Decoder

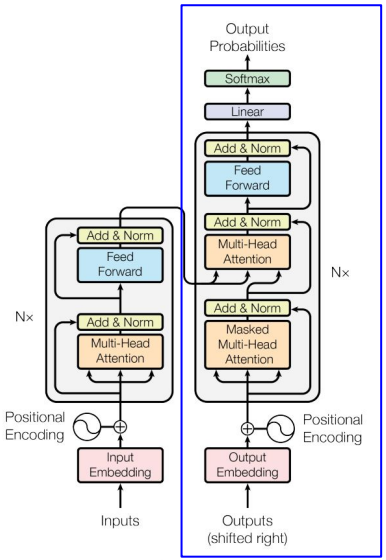


Figure 1: The Transformer - model architecture.

The other part of the transformer is generally referred to as the **decoder**. Its purpose is to **generate the next word in the sequence**

After a long day at work, she sat down to enjoy a warm cup of _____

To achieve this, it relies on the **embeddings processed and updated by the encoder**

Most of the components of the decoder have already been explored in previous discussions

Output embedding

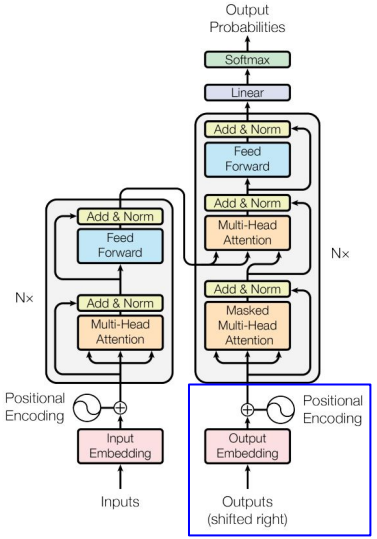
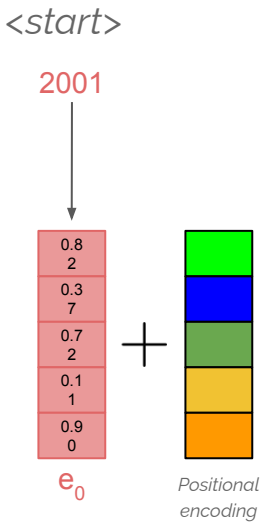


Figure 1: The Transformer - model architecture.

When generating text, we begin by passing the **<start> token** to the decoder, indicating the start of the generation



Multi-Head Attention

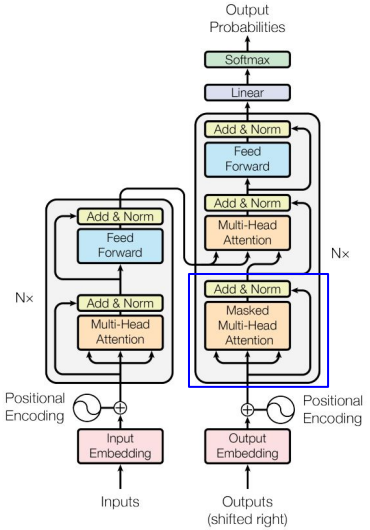
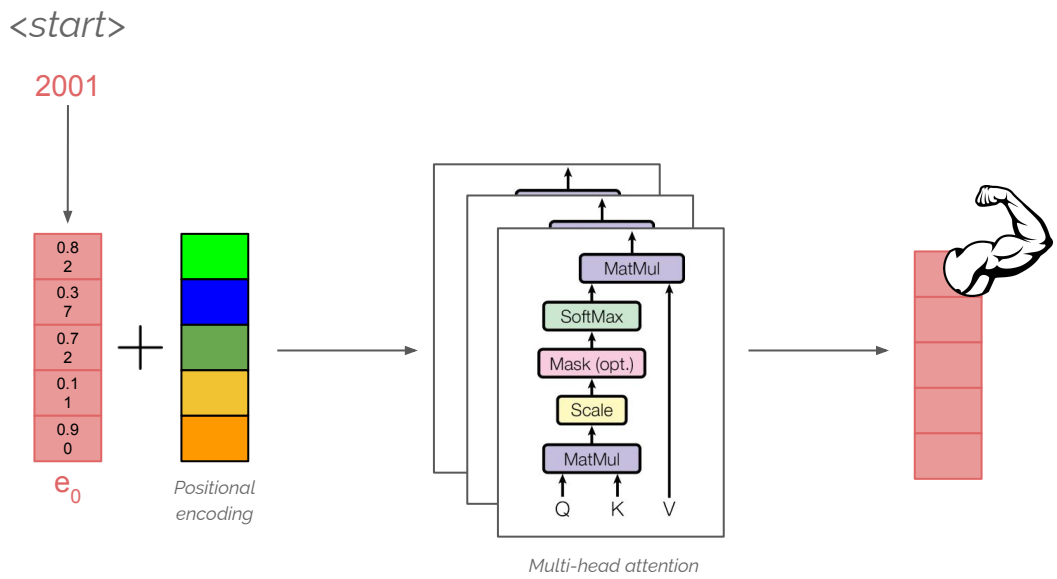


Figure 1: The Transformer - model architecture.

Then, as with the encoder, we apply **multi-head attention** to generate more **complex embeddings** by learning from the context



Integration of the Encoder

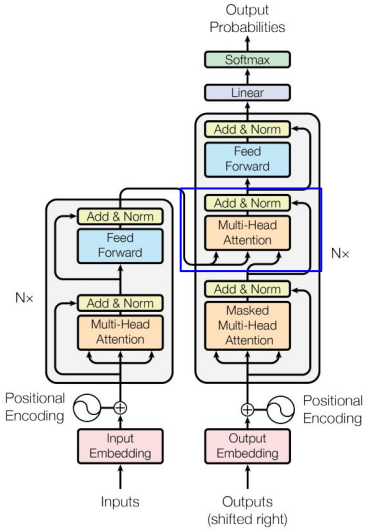
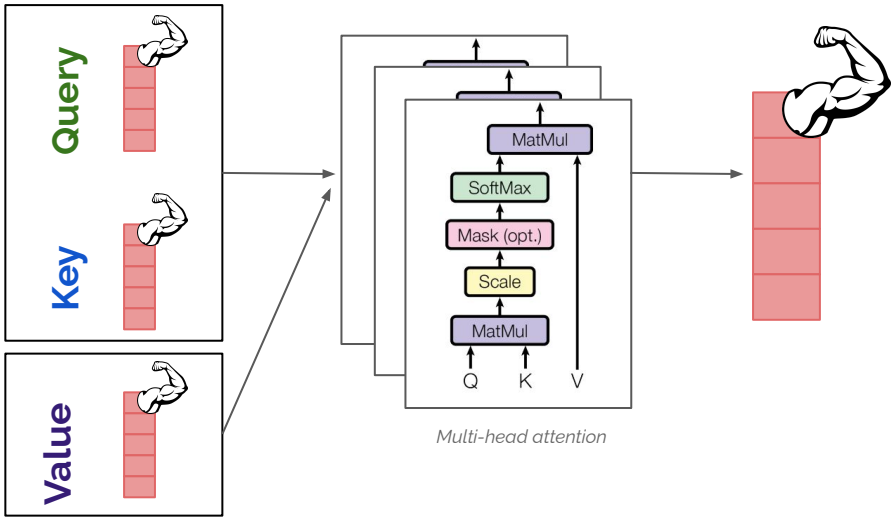


Figure 1: The Transformer - model architecture.

Then, **we combine information from the encoder and the decoder** to take both into account for generating the next token



Multi-head attention

Decoder

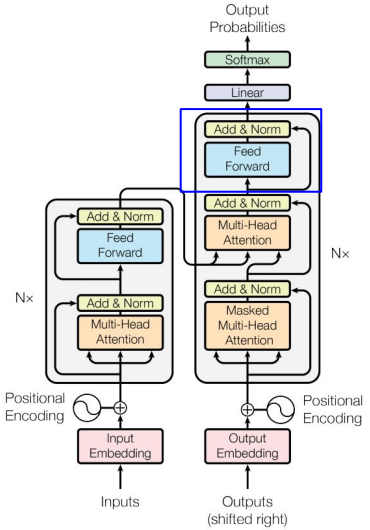
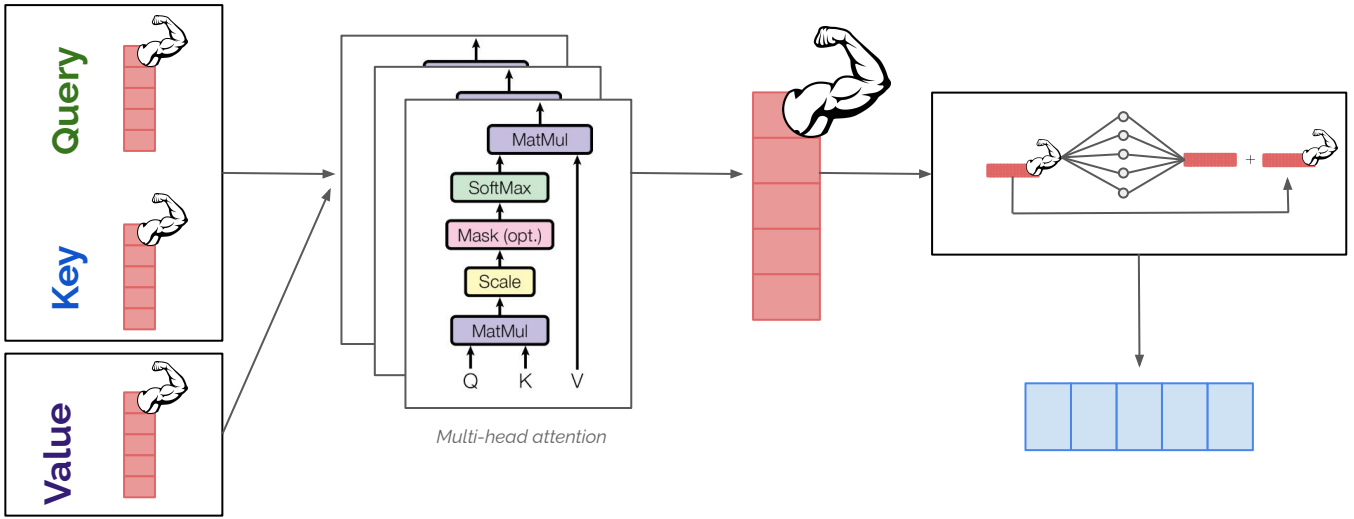


Figure 1: The Transformer - model architecture.

Then, **we combine information from the encoder and the decoder** to take both into account for generating the next token



Final layer

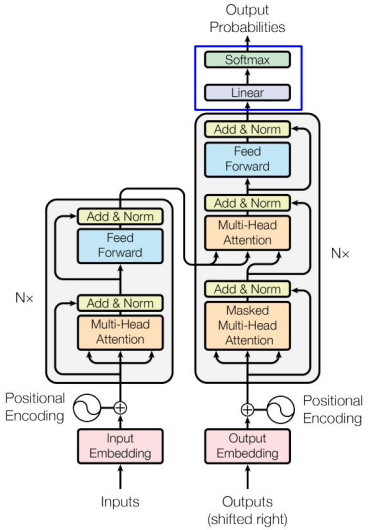
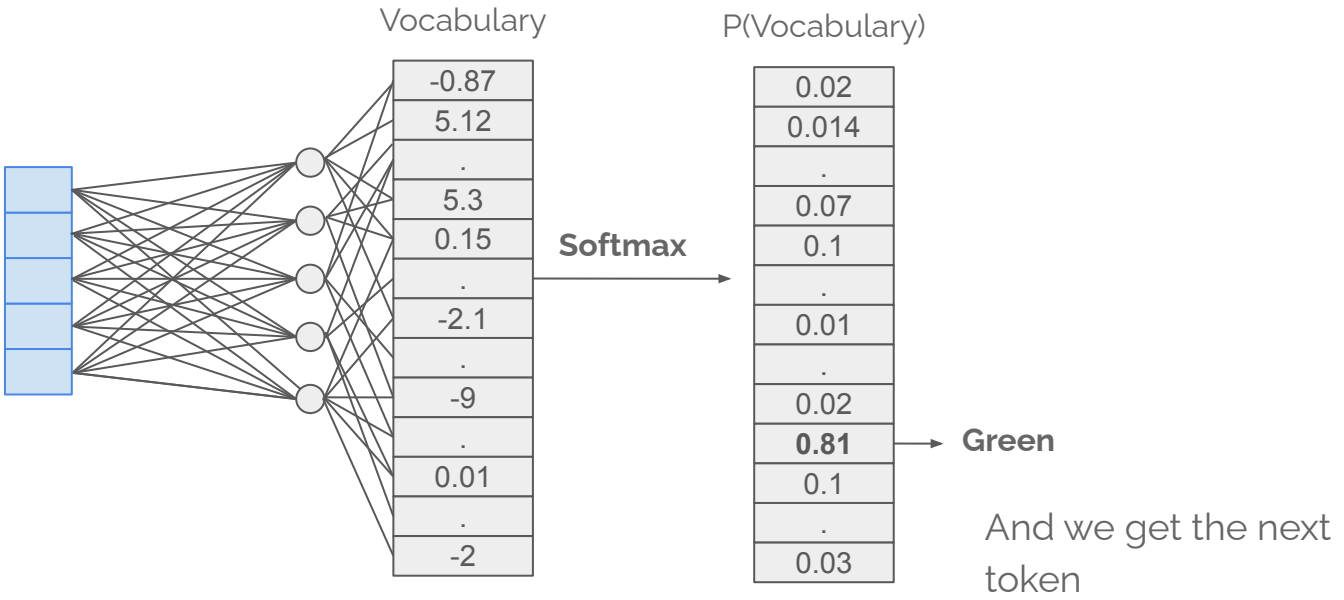


Figure 1: The Transformer - model architecture.

Finally, we apply a **fully-connected layer** to match the shape of the **vocabulary** and use a **softmax** function to generate a **probability distribution** over it



And we get the next token

Token generator

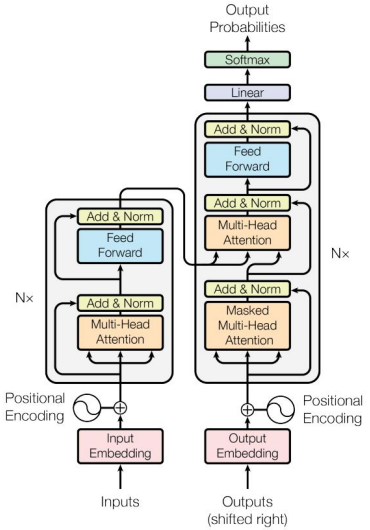
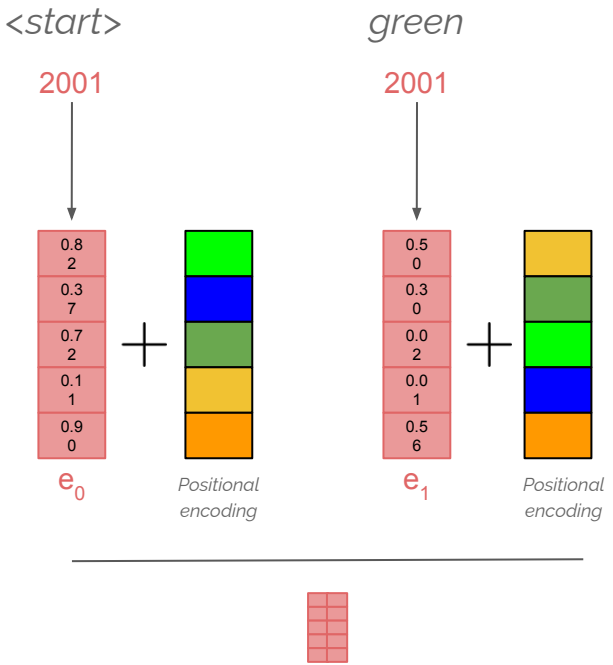


Figure 1: The Transformer - model architecture.

To continue generating tokens, we **add the most recently generated token** and **repeat the process**



Token generator

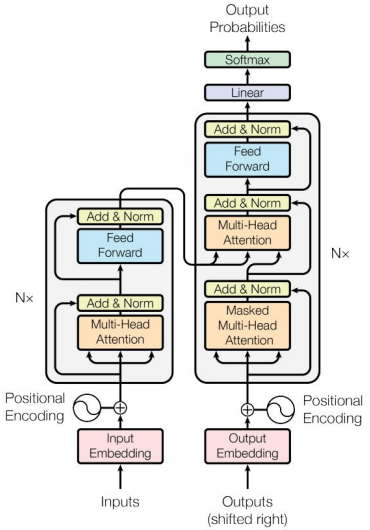
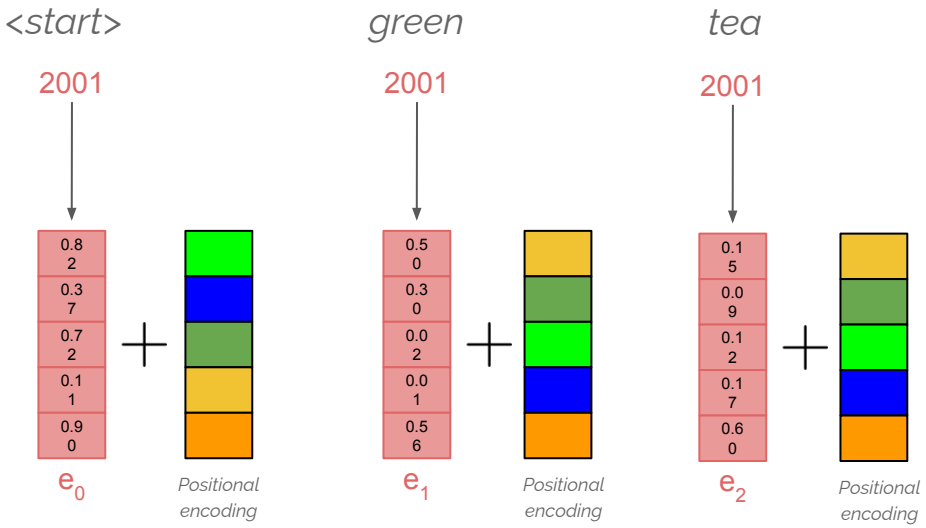


Figure 1: The Transformer - model architecture.

To continue generating tokens, we **add the most recently generated token** and **repeat the process**



Token generator

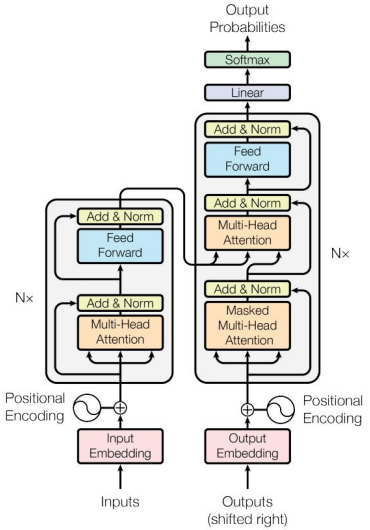
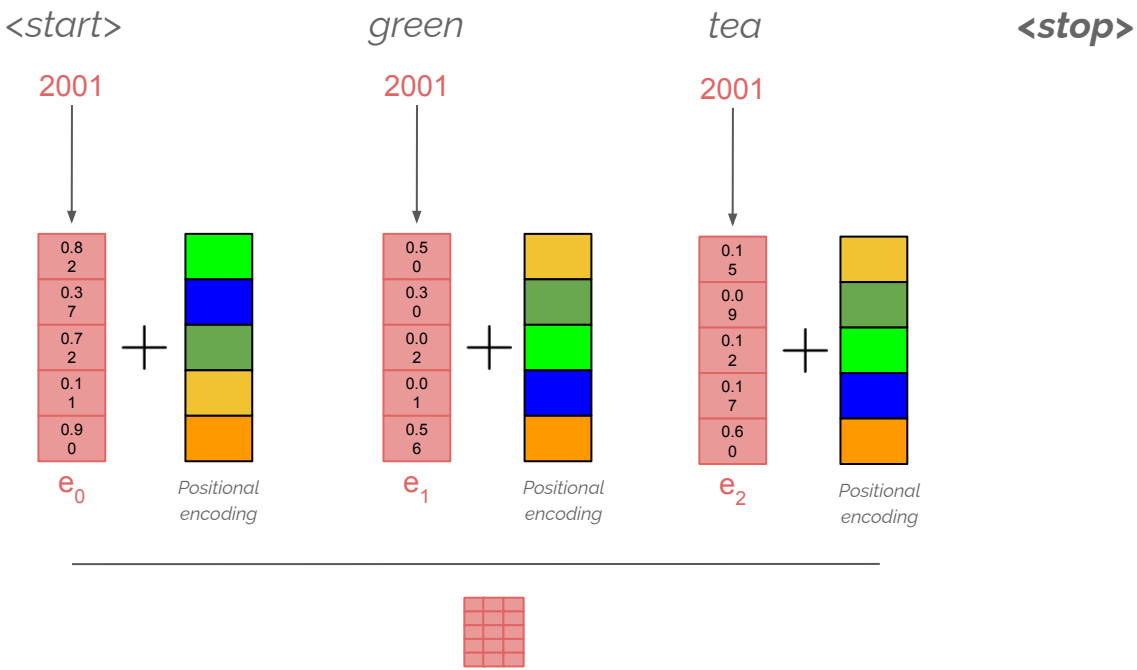


Figure 1: The Transformer - model architecture.

The generation process continues until the transformer produces the **<stop> token**



Encoder and decoder

Encoder and decoder can be used, for instance, for **translation**

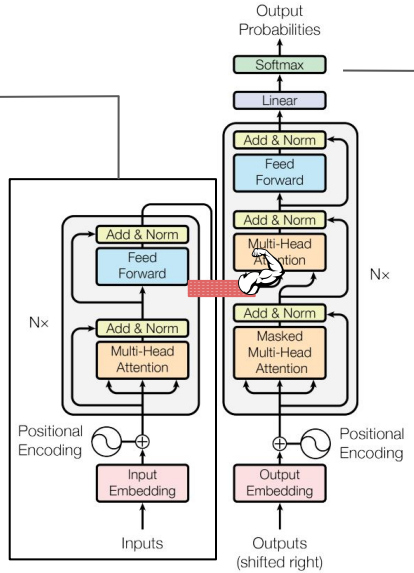


Figure 1: The Transformer - model architecture.

The encoder generates information about the sentence to translate

Este _____

This course is fantastic

Encoder and decoder

But, depending on the task **you might not need an encoder and a decoder**

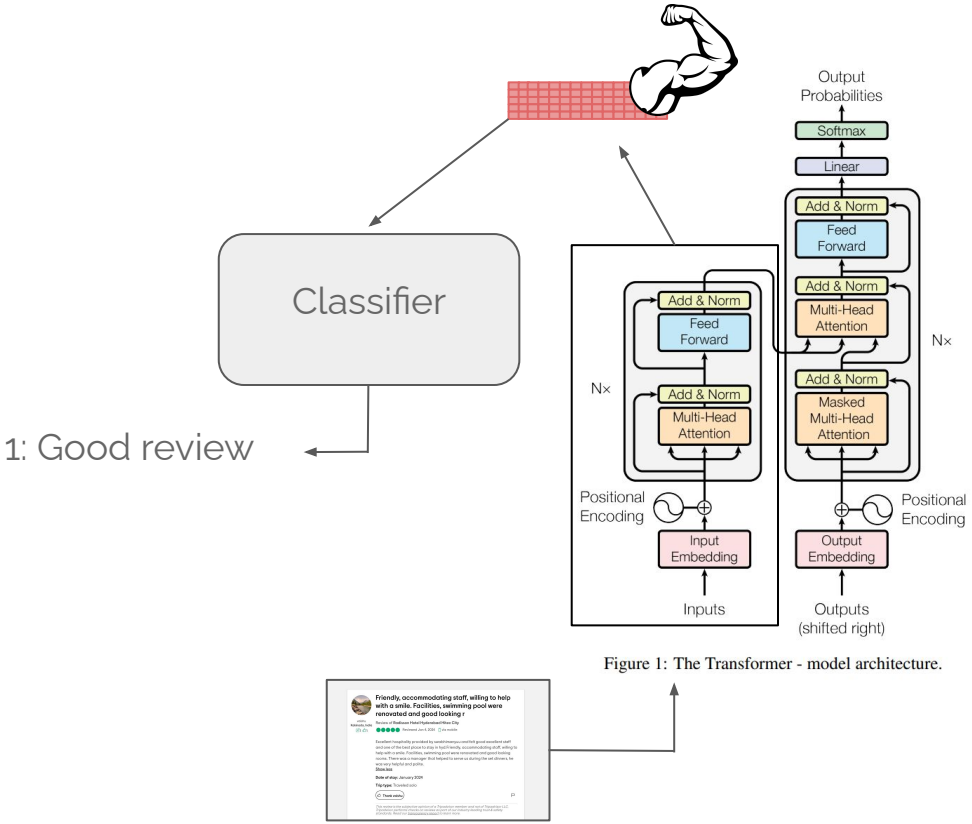


Figure 1: The Transformer - model architecture.

We could train a **classifier** over the embeddings of the **encoding**

Encoder and decoder

But, depending on the task **you might not need an encoder and a decoder**

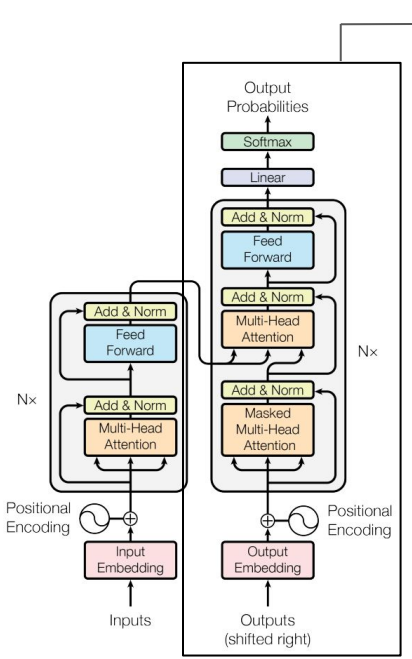


Figure 1: The Transformer - model architecture.

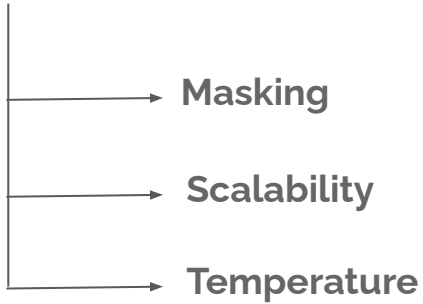
fantastic

We could train a **sequence generator without the context of the encoder**

This course is _____

Additional observations

There are some **observations** worth mentioning regarding transformers



Masking

When training a transformer, we can generate **multiple training samples from a single text input**

After a long day at work, she sat down to enjoy a warm cup of coffee.

After _____

After a _____

After a long _____

After a long day _____

After a long day at _____

After a long day at work _____

After a long day at work, she _____

After a long day at work, she sat _____

After a long day at work, she sat down _____

After a long day at work, she sat down to _____

After a long day at work, she sat down to enjoy _____

After a long day at work, she sat down to enjoy a warm _____

After a long day at work, she sat down to enjoy a warm cup of _____

Masking

When training a transformer, we can generate **multiple training samples from a single text input**

After a long day at work, she sat down to enjoy a warm cup of coffee.

After _____

After a _____

After a long _____

After a long day _____

After a long day at _____

After a long day at work _____

After a long day at work, she _____

After a long day at work, she sat _____

After a long day at work, she sat down _____

After a long day at work, she sat down to _____

After a long day at work, she sat down to enjoy _____

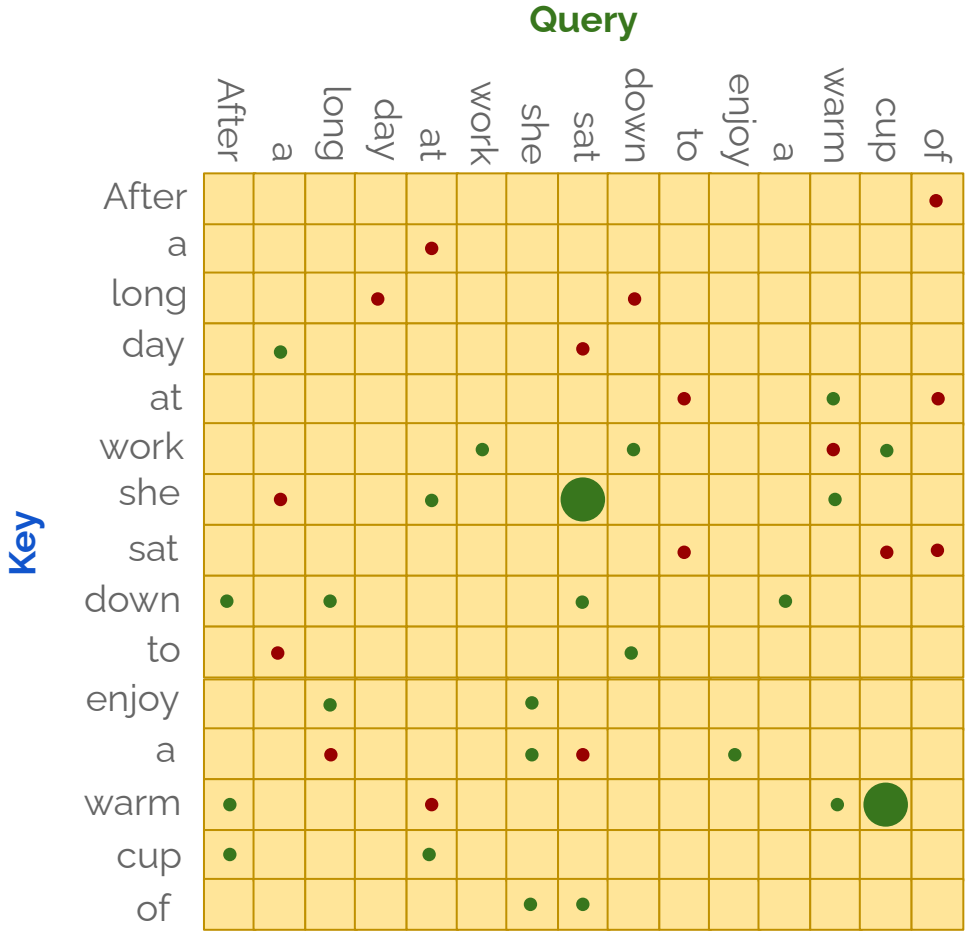
After a long day at work, she sat down to enjoy a warm _____

After a long day at work, she sat down to enjoy a warm cup of _____

How can we **optimize** this process?

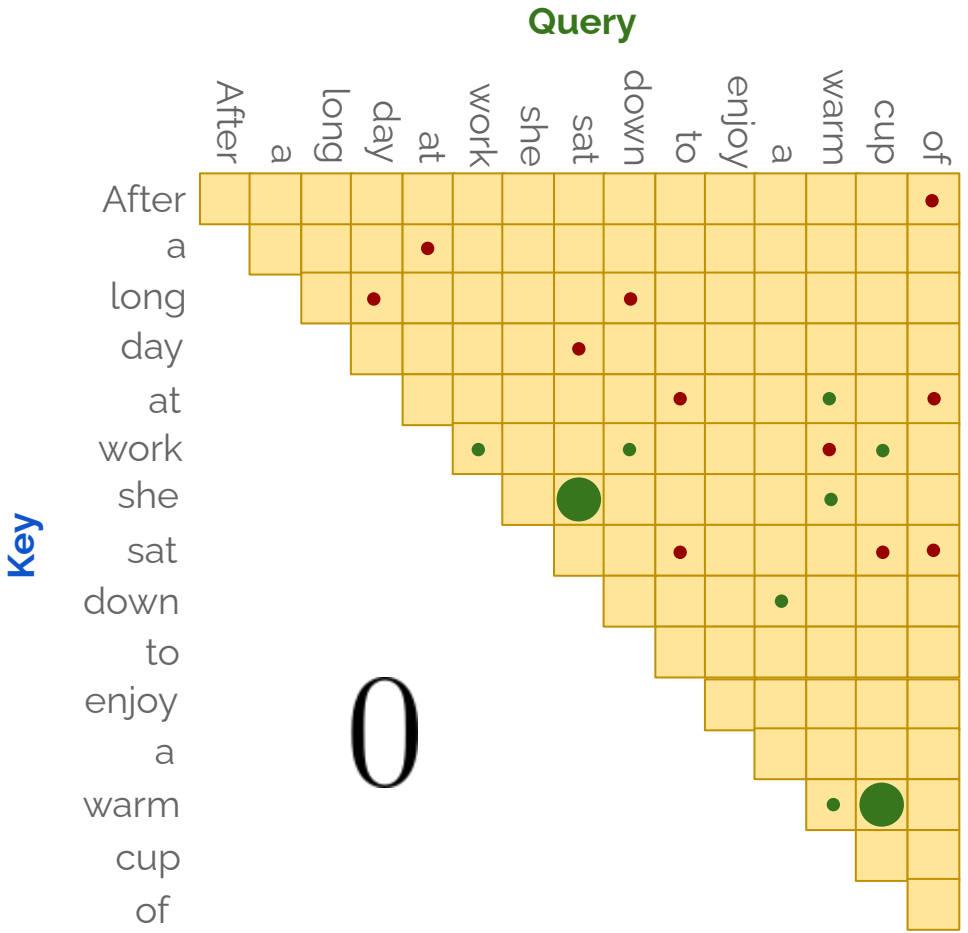
Masking

How can we force the attention mechanism to **only attend to previous words**?



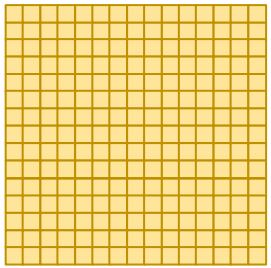
Masking

In this way, the transformer can only update the embeddings **based on the context of previous words**, thus allowing for a **efficient training**



Scalability

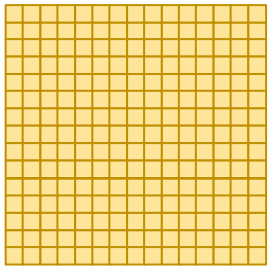
One of the main concerns of transformers concerns the **attention matrix**



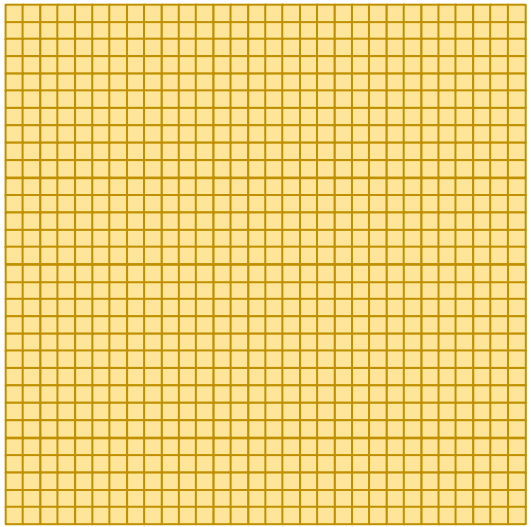
15 tokens

Scalability

One of the main concerns of transformers concerns the **attention matrix**



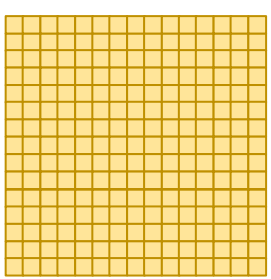
15 tokens



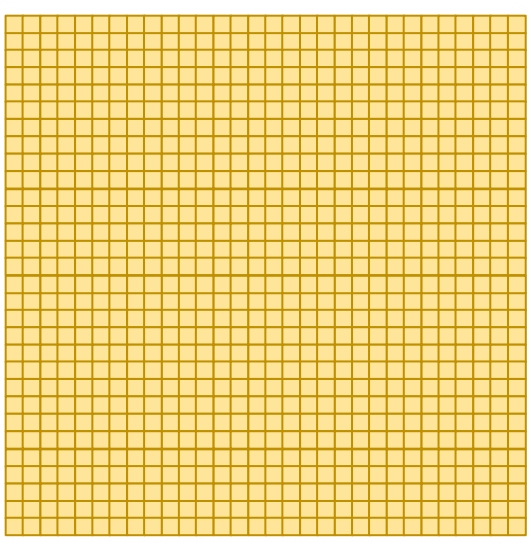
30 tokens

Scalability

One of the main concerns of transformers concerns the **attention matrix**



15 tokens



L

30 tokens

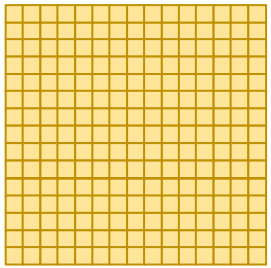
Number of computations: $O(L^2)$

Memory requirement: $O(L^2)$

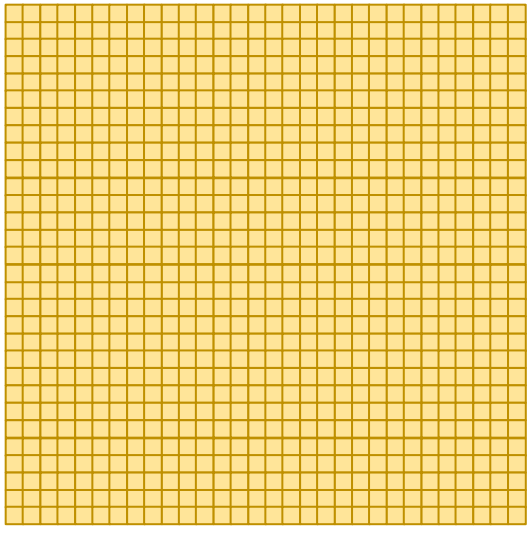
Transformers **do not scale well** with respect to the **size of their context**

Scalability

One of the main concerns of transformers concerns the **attention matrix**



15 tokens



L

30 tokens

Number of computations: $O(L^2)$

Memory requirement: $O(L^2)$

Transformers **do not scale well** with respect to the **size of their context**

However, there exist solutions

$L \sim 10^6$

Temperature

After a long day at work, she sat down to enjoy a warm cup of _____

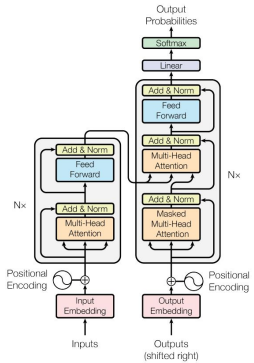


Figure 1: The Transformer - model architecture.

P(Vocabulary)

0.02
0.014
.
0.07
0.1
.
0.01
.
0.02
0.81 → **Green**
0.1
.
0.03

How can we encourage more **creativity** in the model?

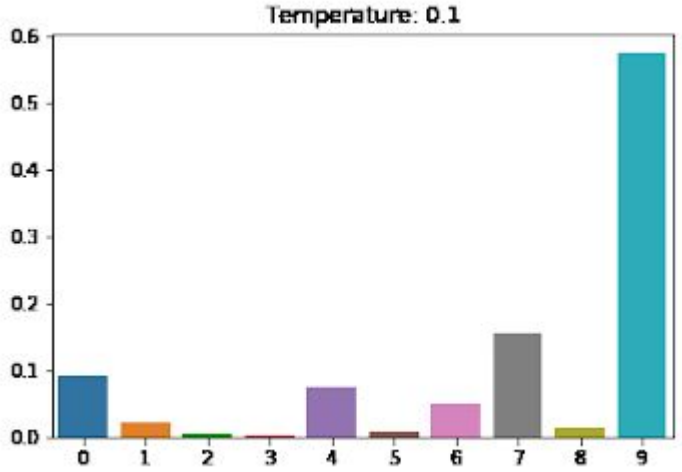
$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \xrightarrow{\text{Temperature}} \text{softmax}(z)_i = \frac{e^{z_i/T}}{\sum_{j=1}^{N} e^{z_j/T}}$$

$T > 1$ → Creativity

$T = 1$

$T < 1$ → Coherence

Temperature



$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \xrightarrow{\text{Temperature}} \text{softmax}(z)_i = \frac{e^{z_i/T}}{\sum_{j=1}^{N} e^{z_j/T}}$$

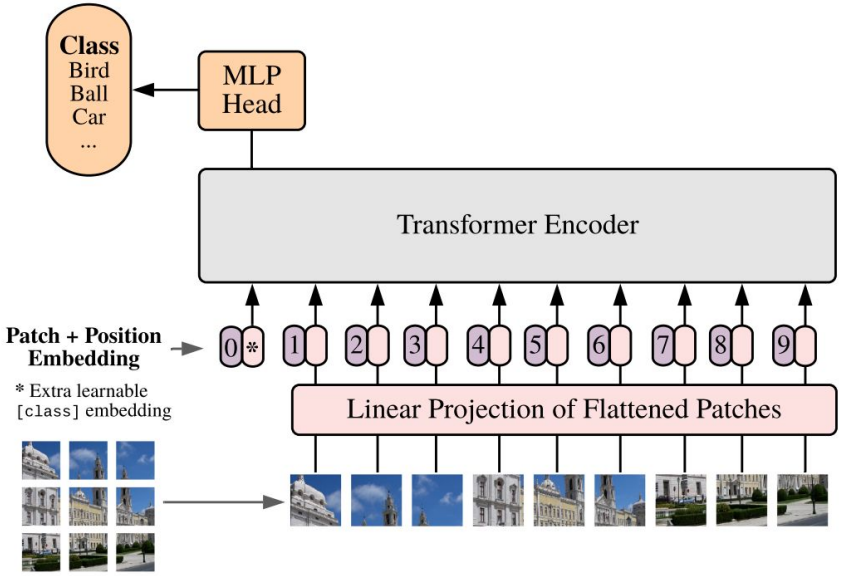
$T > 1 \longrightarrow$ Creativity

$T = 1$

$T < 1 \longrightarrow$ Coherence

Beyond NLP - Vision Transformers

- Vision Transformers (ViT) is the application of Transformers for image classification
- The ViT team knew they couldn't exactly mimic the language approach since self-attention on every pixel would be prohibitively expensive in computing time.
- Instead, they **divided the larger image into square units, or tokens**. The size is arbitrary, as the tokens could be made larger or smaller depending on the resolution of the original image (the default is 16 pixels on a side).



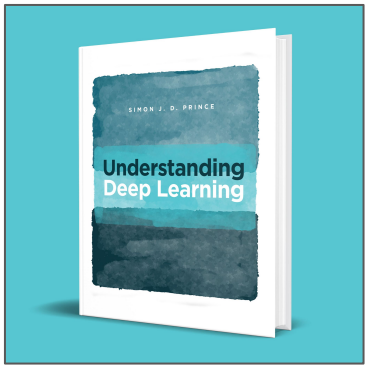
CNN vs **ViT**

- In CNNs, you start off being very local and slowly get a global perspective. But in transformers, with self-attention, even the very first layer of information processing makes connections between distant image locations.
- CNNs assume that nearby pixels are related (locality) and that different parts of an image are processed similarly (weight sharing) (*inductive bias*). These assumptions help CNNs learn effectively with limited training data. But this means, that given enough data, ViTs will learn better global relationships (because they don't assume locality).
- CNNs are smaller, so they are preferred for applications that need high latency

Resources

Video resources

- 3Blue1Brown series
- Karpathy - Deep Dive into LLMs like ChatGPT
- Ari Seff - What are Transformer Neural Networks?



Text resources

- Understanding Deep Learning - Simon Prince

Code resources

- Minimal implementation of GPT (PyTorch)

Questions

Jose González-Abad
Instituto de Física de Cantabria (CSIC-UC)


gonzabad@ifca.unican.es


jgonzalezab


Image sources