Attention Mechanism and Transformers

Video 7: Introduction to Transformers

Analytics Vidhya

IN AIR

Analytics Vidhya

ChatGPT

Analytics Vidhya

Transformers

Purpose: Addresses sequence-to-sequence issues with long-range dependencies.

Impact: Integral to NLP, driving advancements like BERT, GPT 2, T5, Chat GPT.

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

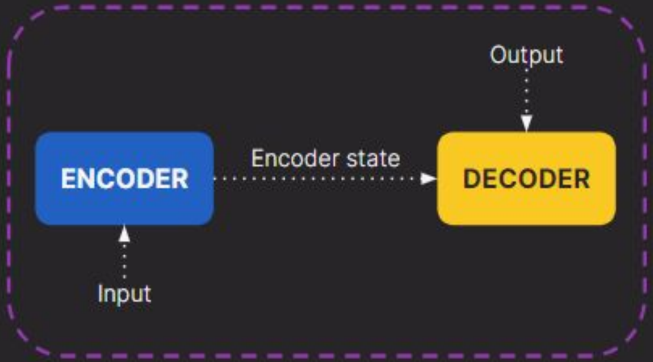
Illia Polosukhin[* ‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Recap

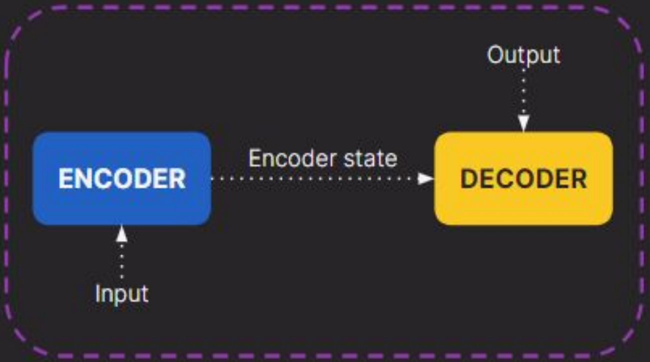
Encoder - Decoder



Encoder state : solely responsible to transfer information to decoder.

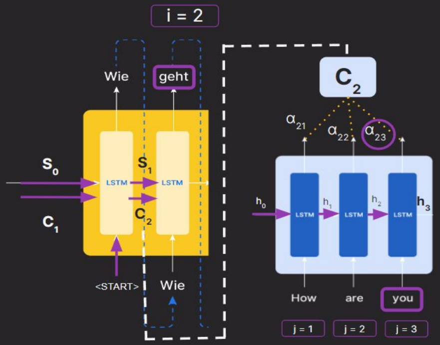
Recap

Encoder - Decoder



Encoder state : solely responsible to transfer information to decoder.

Attention Mechanism



$$C_i = \sum \alpha_{ij} h_i$$

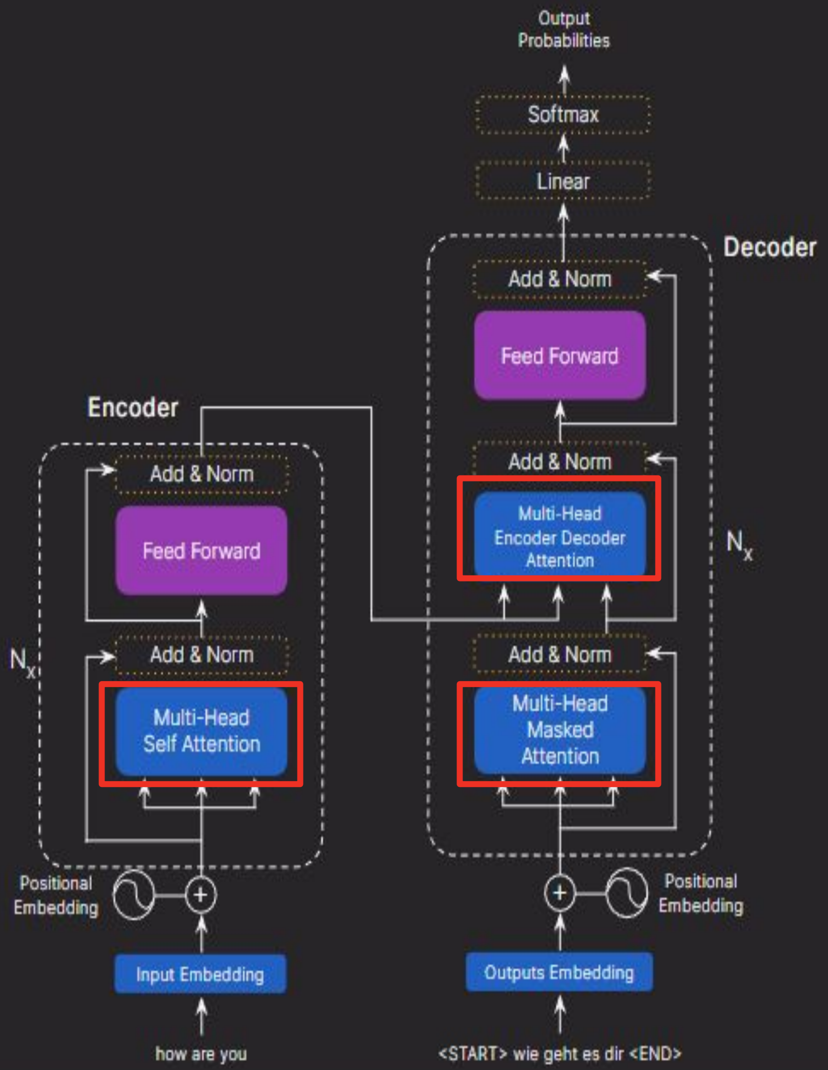
Calculation of alpha **increase computation** for long documents.

Encoder processes the **input sequentially**

Architecture

Attention

- Encoder-Decoder Attention
- Self Attention
- Masked Attention
- Multi-Head Attention



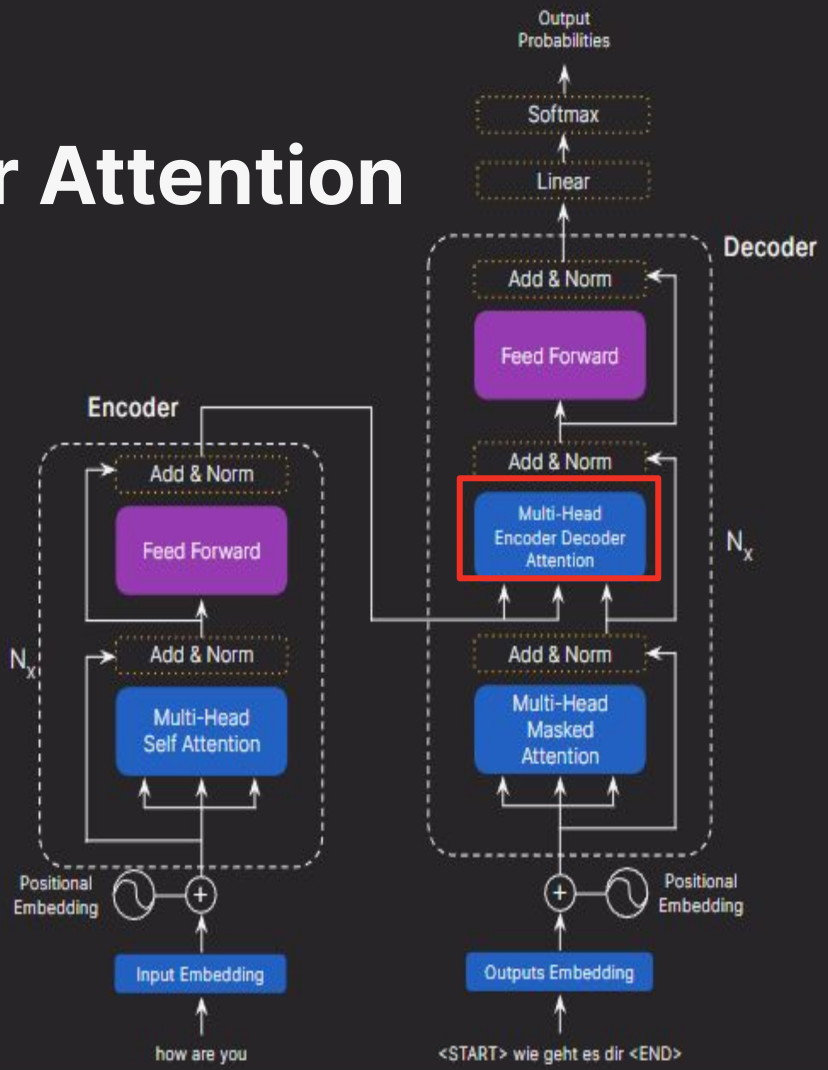
Encoder - Decoder Attention

Encoder- Decoder Attention

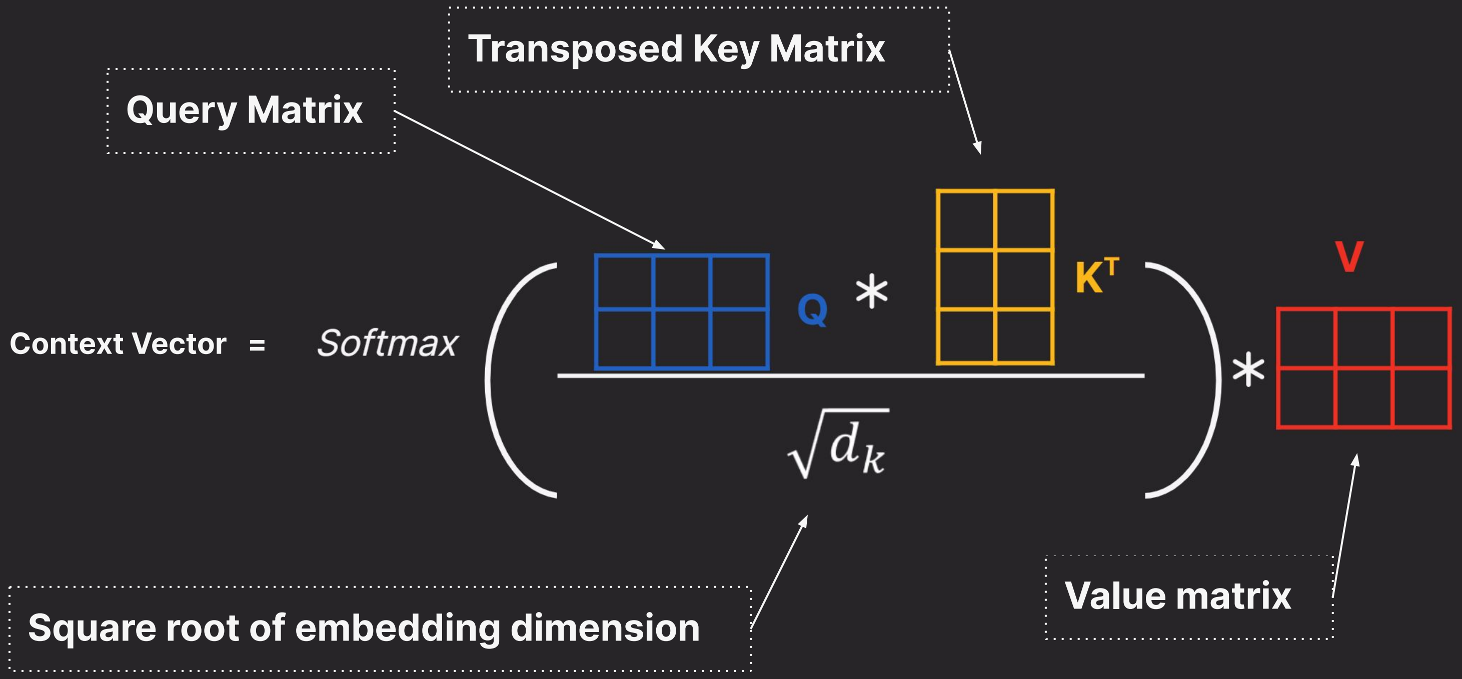
$$C_i = \sum \alpha_{ij} h_i$$

Uses Dot-Product attention to calculate attention.

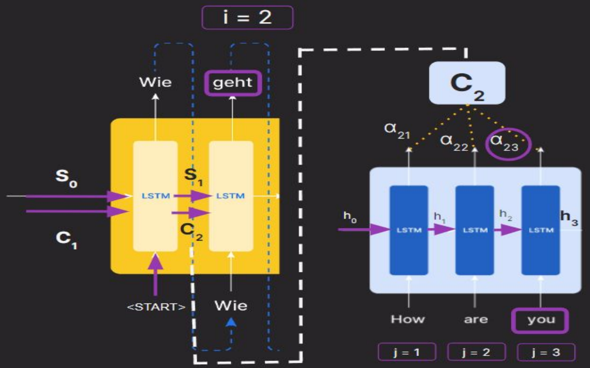
$$\text{Context Vector} = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d^k}}\right) * V$$



Dot Product Attention



Dot Product Attention



$$C_i = \sum \alpha_{ij} h_i$$

$$\textbf{Context Vector} = \textit{Softmax}\left(\underbrace{\frac{Q * K^T}{\sqrt{d_k}}}_{\sum \alpha_{ij}}\right) * \underbrace{V}_{h_j} = Z$$

Attention weights .... $\sum \alpha_{ij}$

h_j .... Hidden States

Dot Product Attention

Q **Query Vector** : Importance of Relative Words

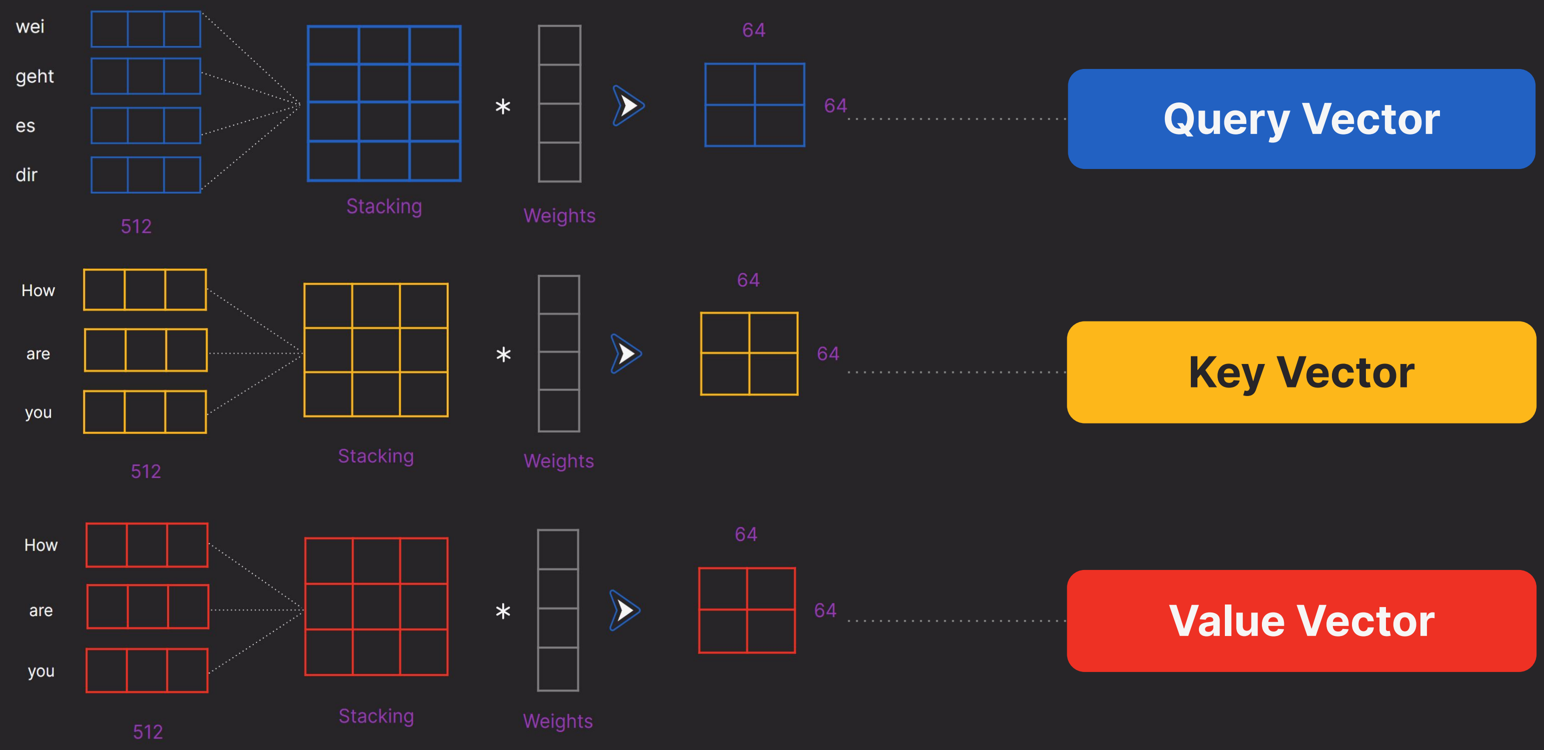
K **Key Vector** : Represents Evaluated Word

V **Value Vector** : Contains focussed information

Translate English to German:

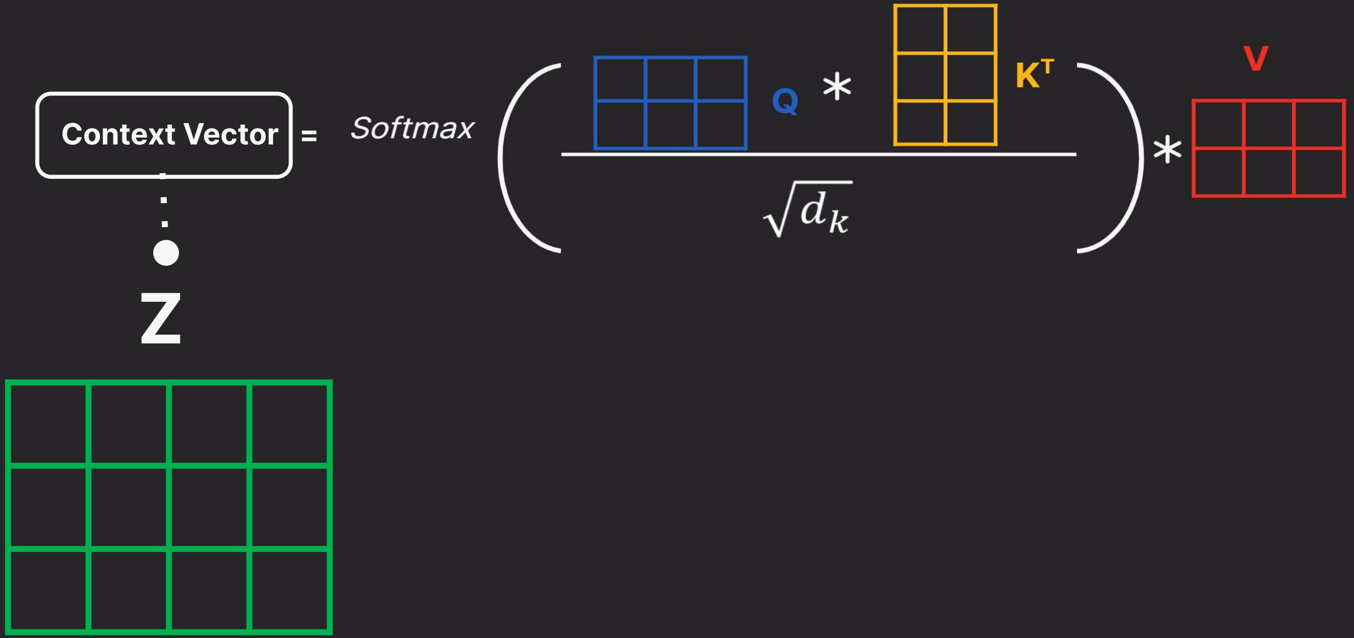
"How are you?" → "Wie geht es dir?"

Encoder-Decoder Dot Product Attention

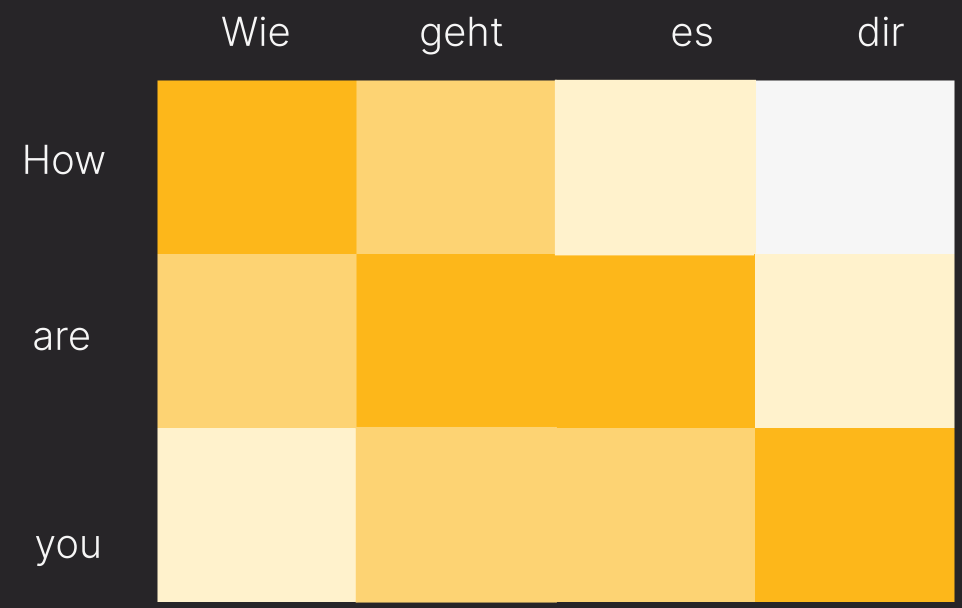


Encoder-Decoder Dot Product Attention

$$Cv = \text{Softmax}\left(\frac{Q \,.\, K^T}{\sqrt{d^k}} \right) * V$$



Encoder-Decoder Dot Product Attention



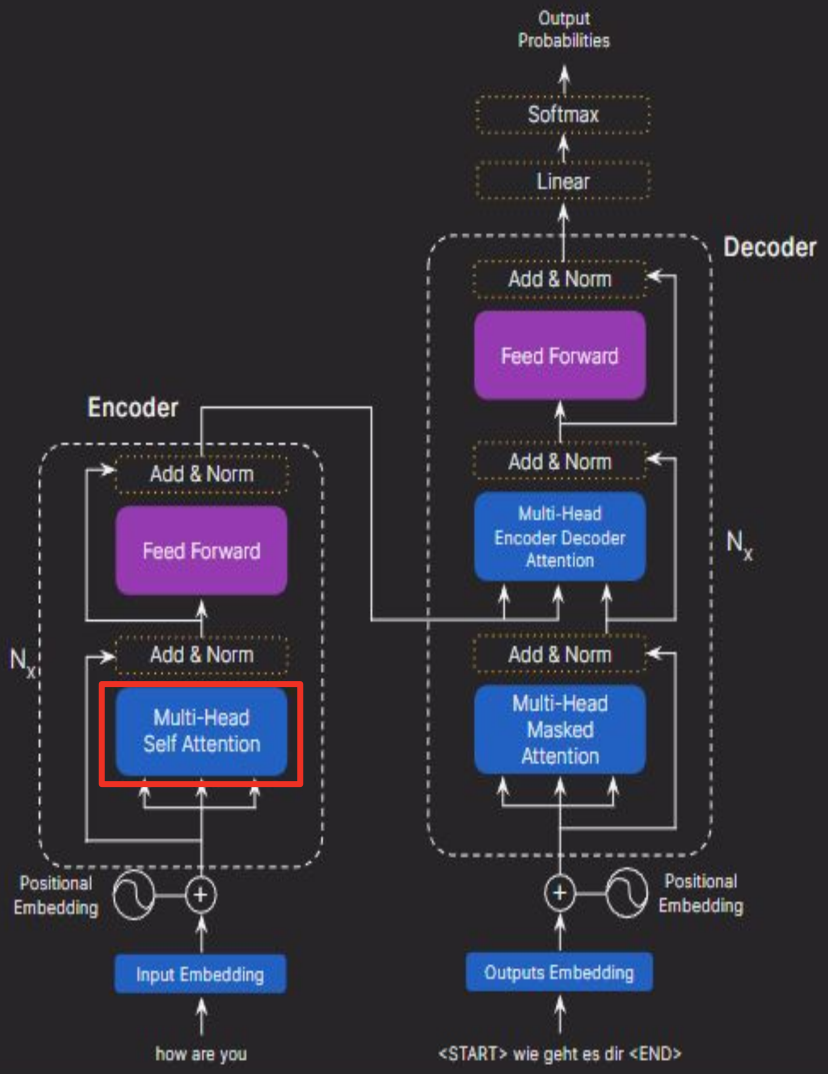
Attention

Encoder-Decoder Attention

Self Attention

Masked Attention

Multi-Head Attention

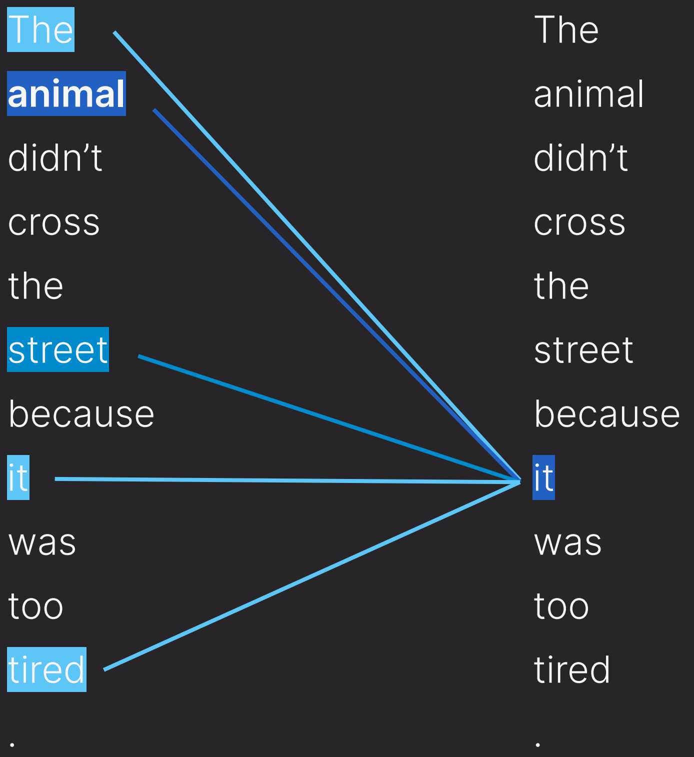


Self Attention

- The animal didn't cross the street because **it** was too tired.
- The animal didn't cross the street because **it** was too crowded.

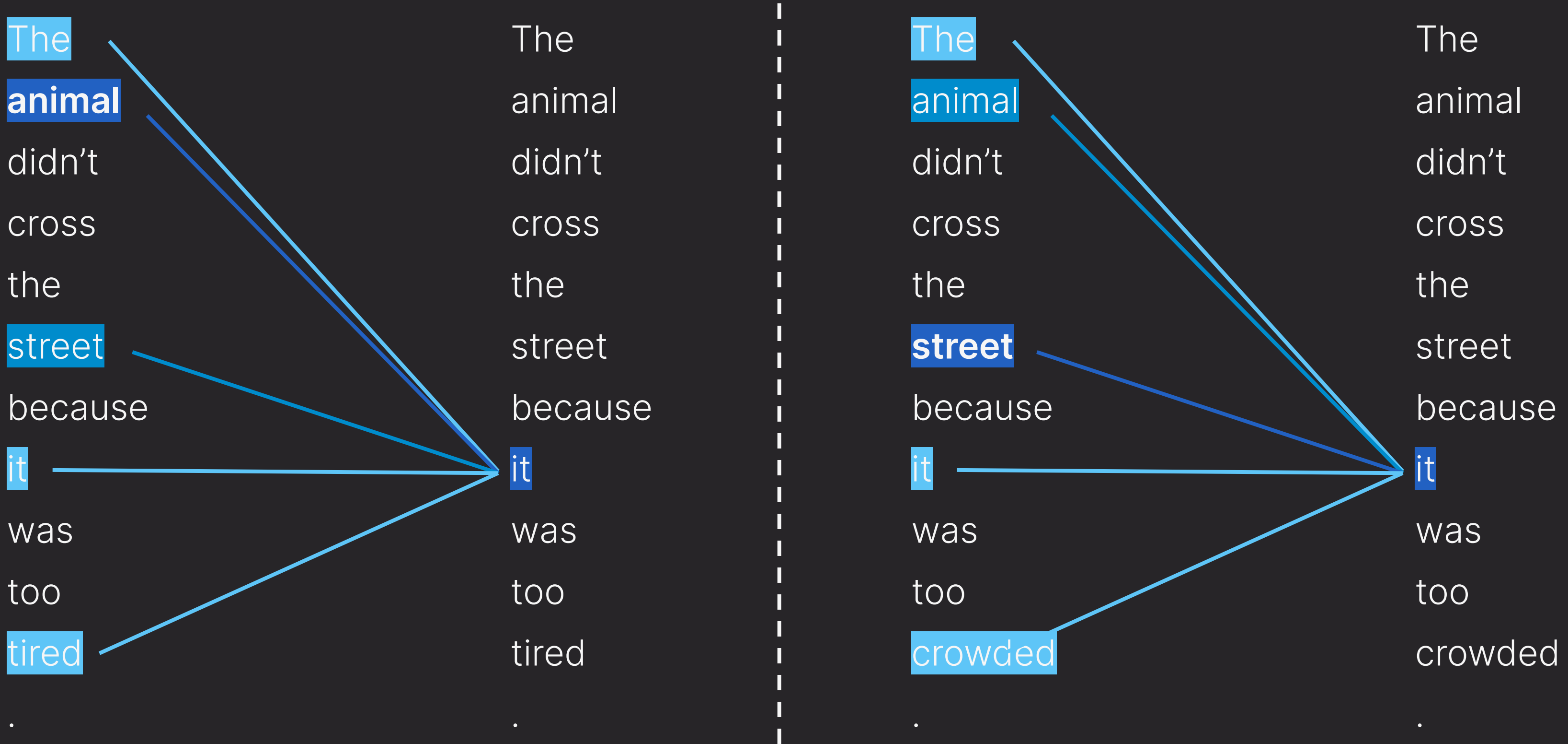
What does “it” refer to in these sentences?

Self Attention



The animal didn't cross the street because **it** was too tired.

Self Attention



The animal didn't cross the street because **it** was too tired.

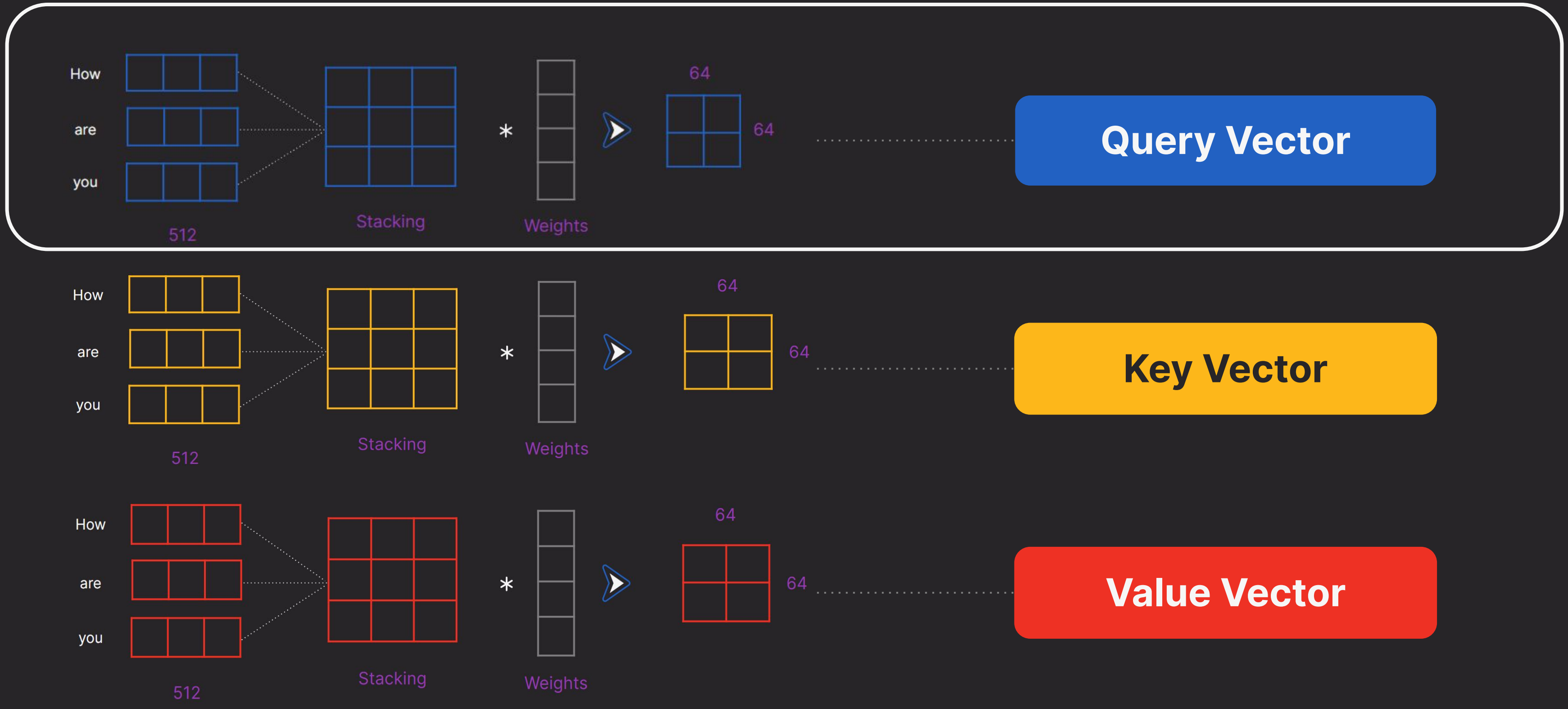
The animal didn't cross the street because **it** was too wide.

Self Attention

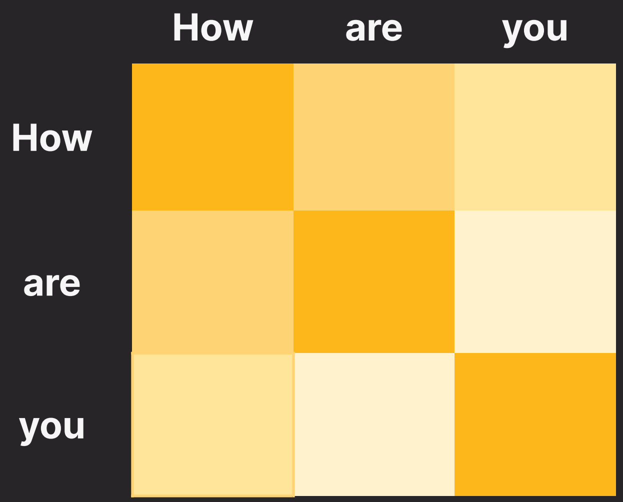
$$C_v = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d^k}}\right) * V$$

Q = Query Vector
K = Key vector
V = Value vector
d^k= dimension of key vector

Self Attention



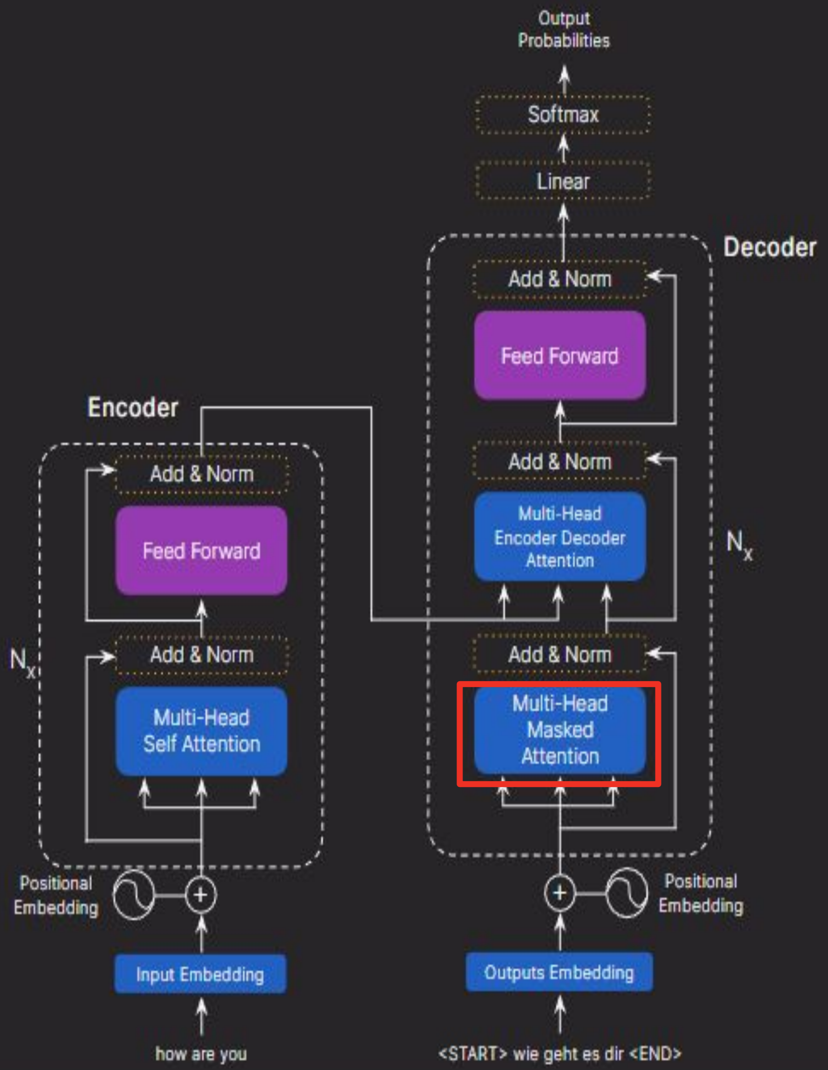
Self Attention



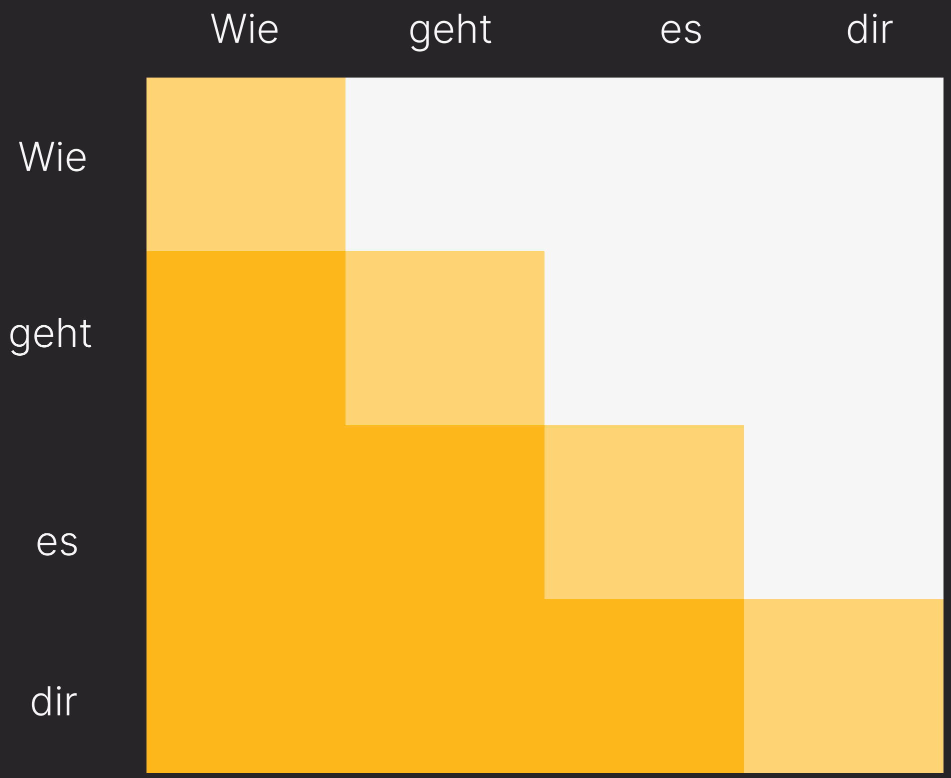
Architecture

Attention

- Encoder-Decoder Attention
- Self Attention
- **Masked Attention**
- Multi-Head Attention



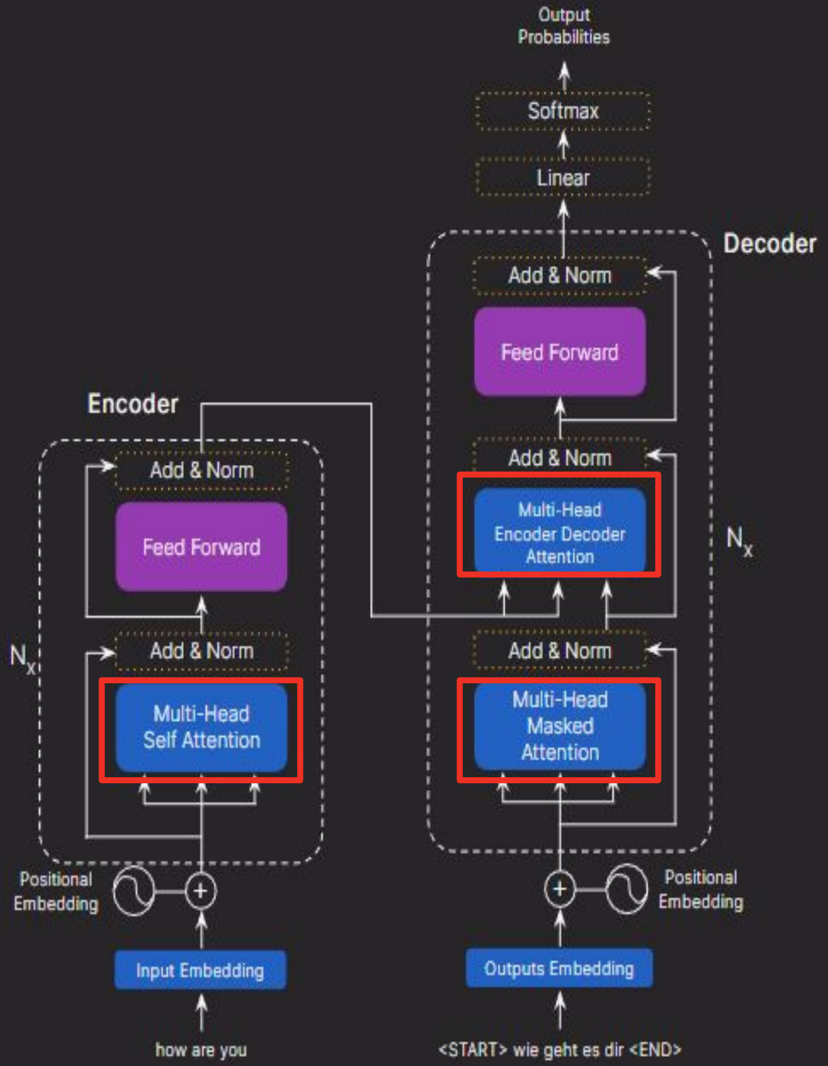
Masked Attention



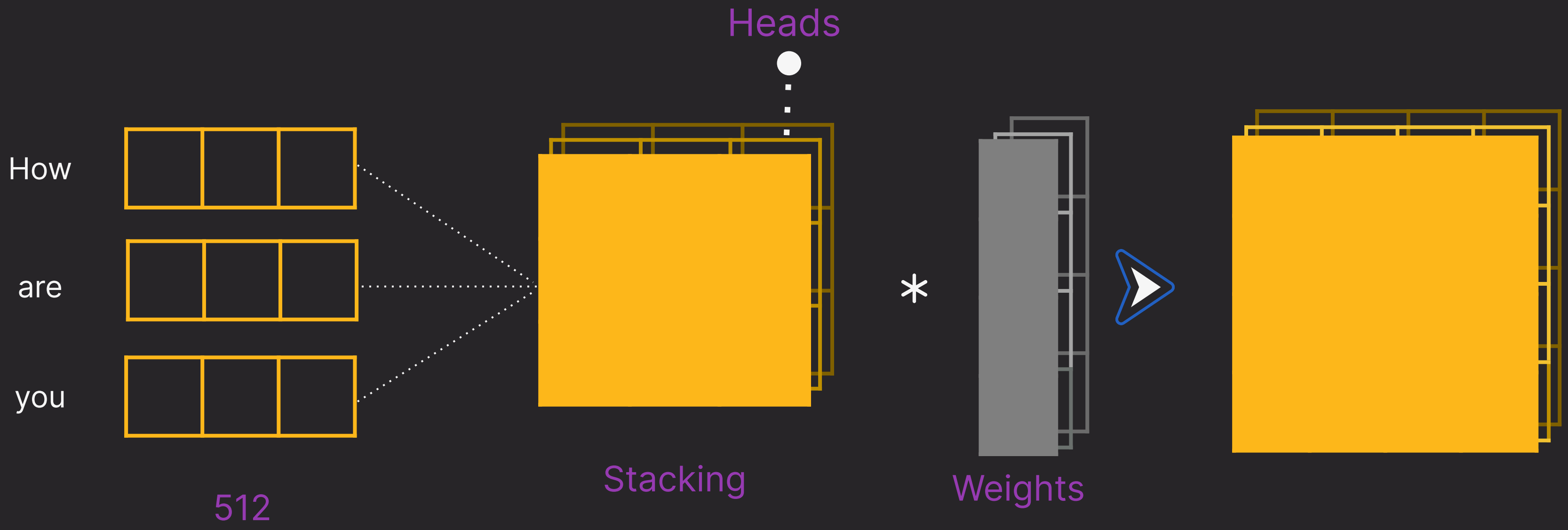
Architecture

Attention

- Encoder-Decoder Attention
- Self Attention
- Masked Attention
- **Multi-Head Attention**



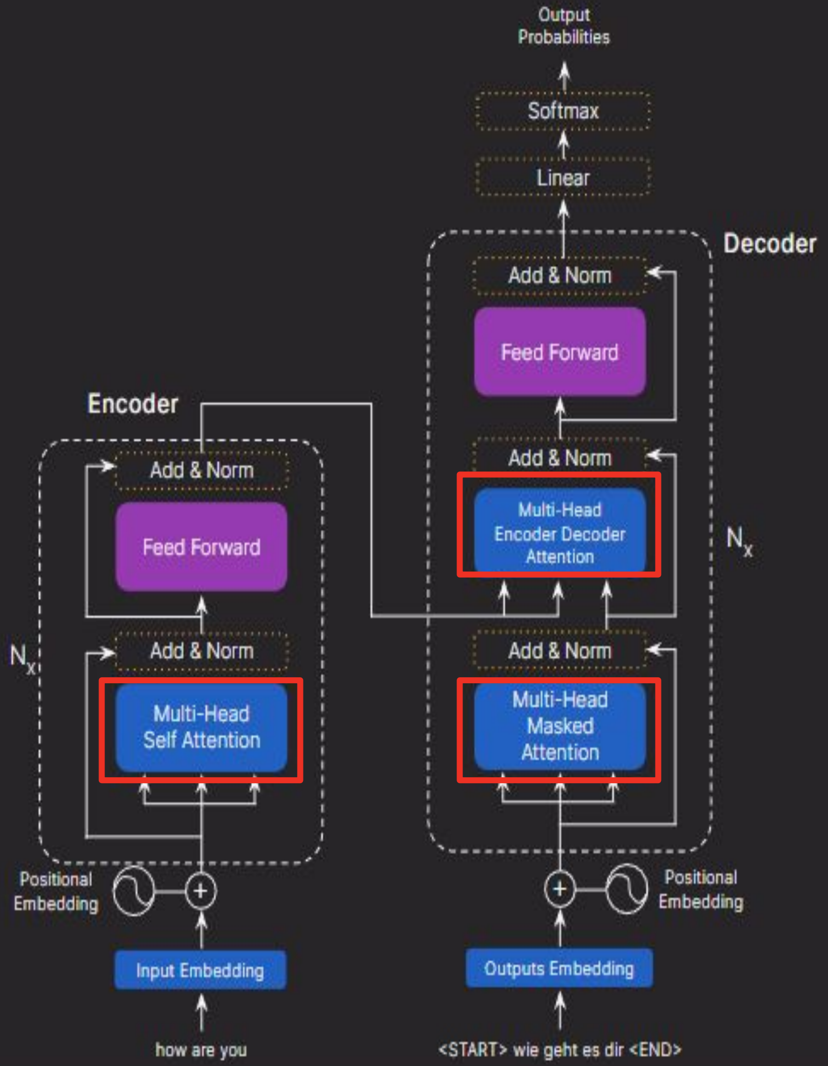
Multi-Head Attention



Architecture

Attention

- Encoder-Decoder Attention
- Self Attention
- Masked Attention
- Multi-Head Attention



Analytics Vidhya