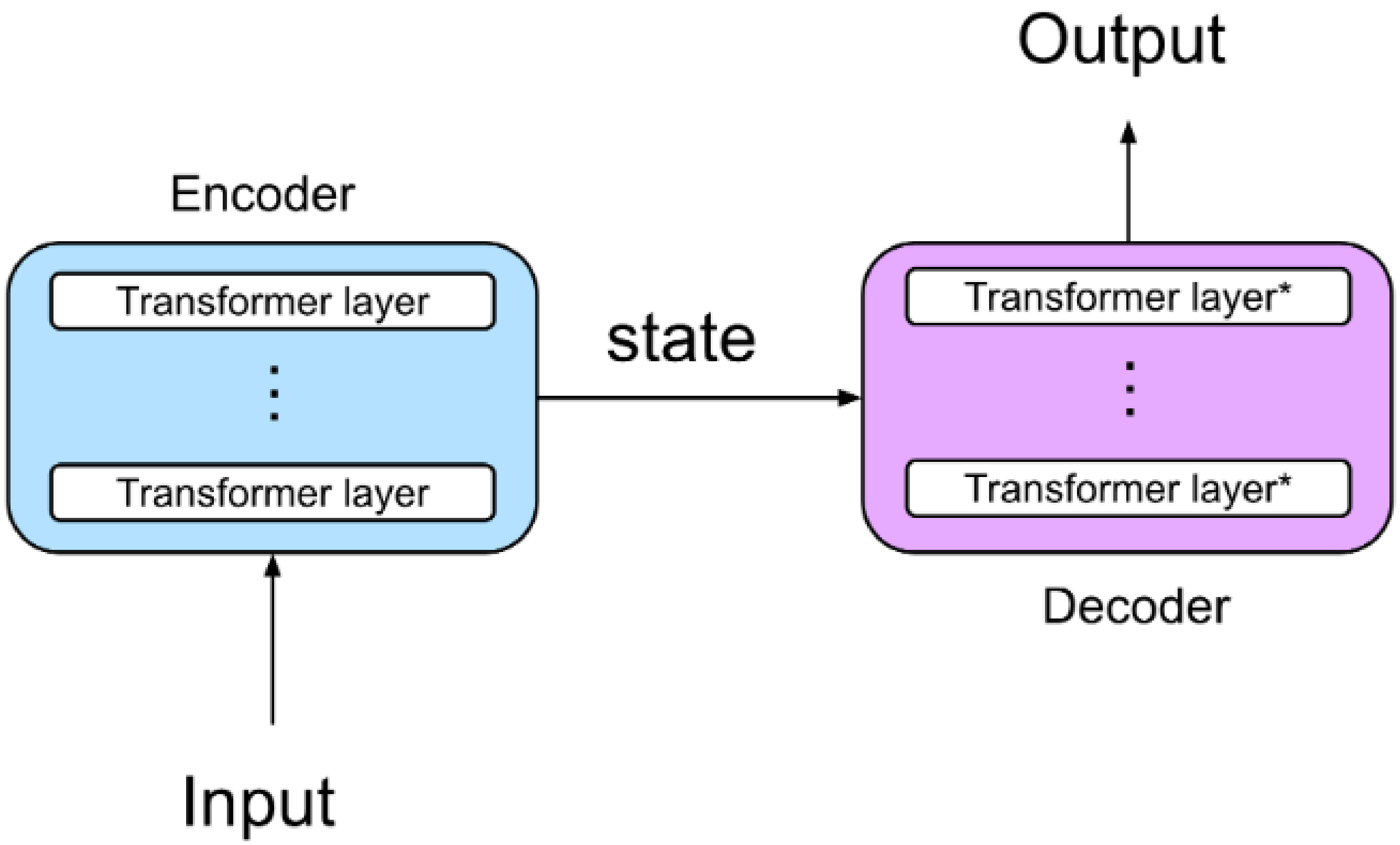
Attention is All You Need

Authored by: Vaswani et al., 2017

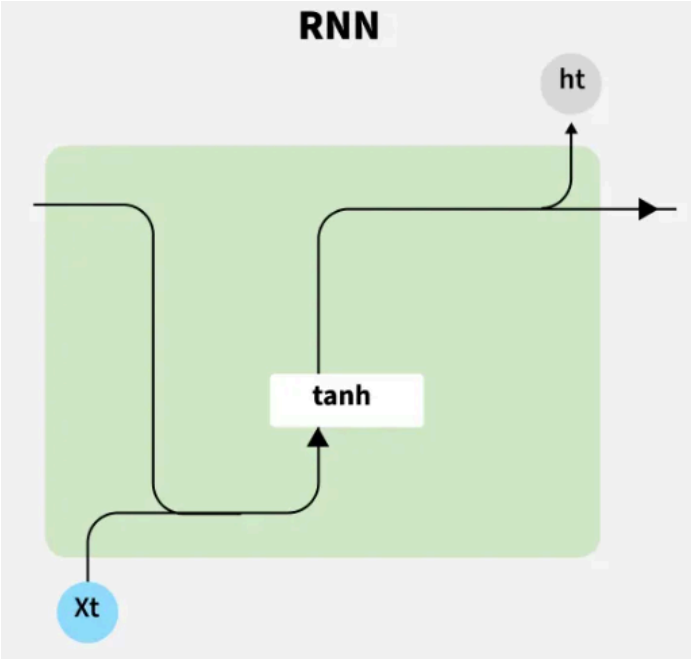


Diagrams: L, J.S. (2023) Intro to Transformer Architecture. https://www.linkedin.com/pulse/intro-transformer-architecture-jithin-s-l/.

INTRODUCTION

- Machine Learning models that take sequence of data as input or produce them as output are known as **sequence models**.
- Sequential data encompasses text steams, audio clips, video clips, time-series data, etc.
- Example of Sequence Models in real-world scenarios:
    i. Speech Recognition
    ii. Sentiment Classification
    iii. DNA sequence analysis
    iv. **Machine Translation: We pass a sentence in one language, like English and the model translates it into a sentence in another language like German.**

TRADITIONAL APPROACHES

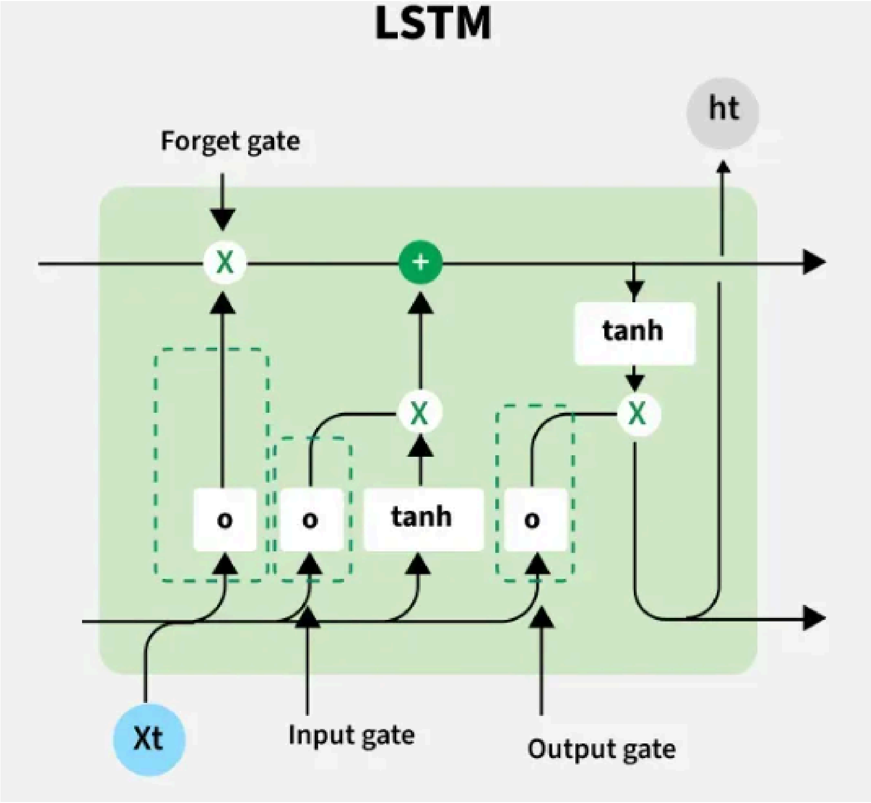


Recurrent Neural Network

- Loops that allow them to retain information from earlier steps, in contrast to standard feedforward networks
- Due to their limited memory, RNNs are not very good at capturing long-range dependencies, but they do well on short sequences.

Long Short-term Memory

- Improved version of RNNs with gating system which makes it possible for LSTM to selectively recall and forget information, which impoves their ability to learn long-term dependencies.



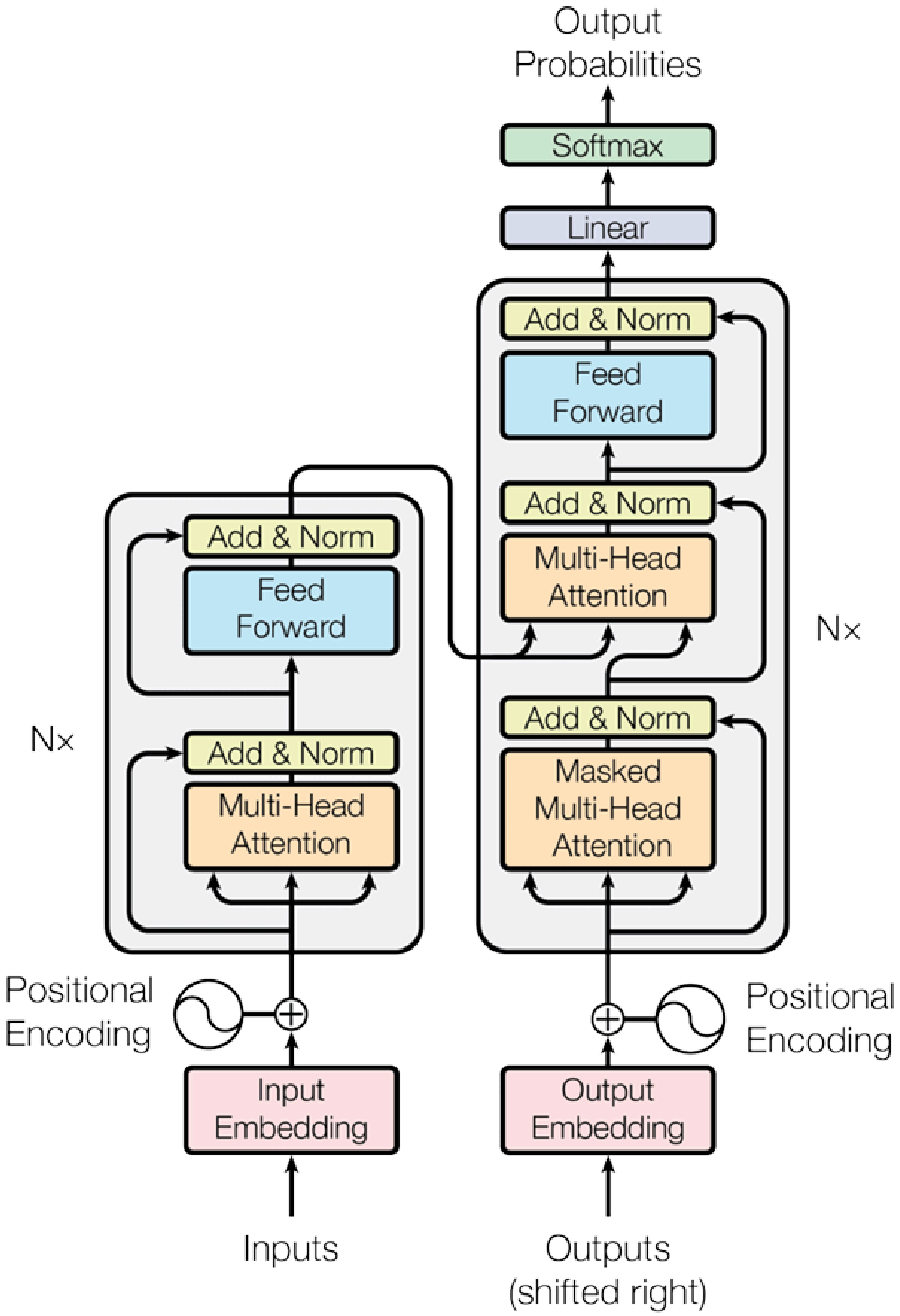
Gated Recurrent Neural Network

- GRUs combine input and forget gates into a single update gate.
- They still rely on sequential processing which limits parallaelization and slows training on long sequences.

Diagrams: GeeksforGeeks (2025) RNN vs LSTM vs GRU vs Transformers. Available at: https://www.geeksforgeeks.org/rnn-vs-lstm-vs-gru-vs-transformers/

MODEL ARCHITECTURE

- The transformer follows an **encoder-decoder** design.
- Both encoder and decoder are composed of stacked layers (typically 6).



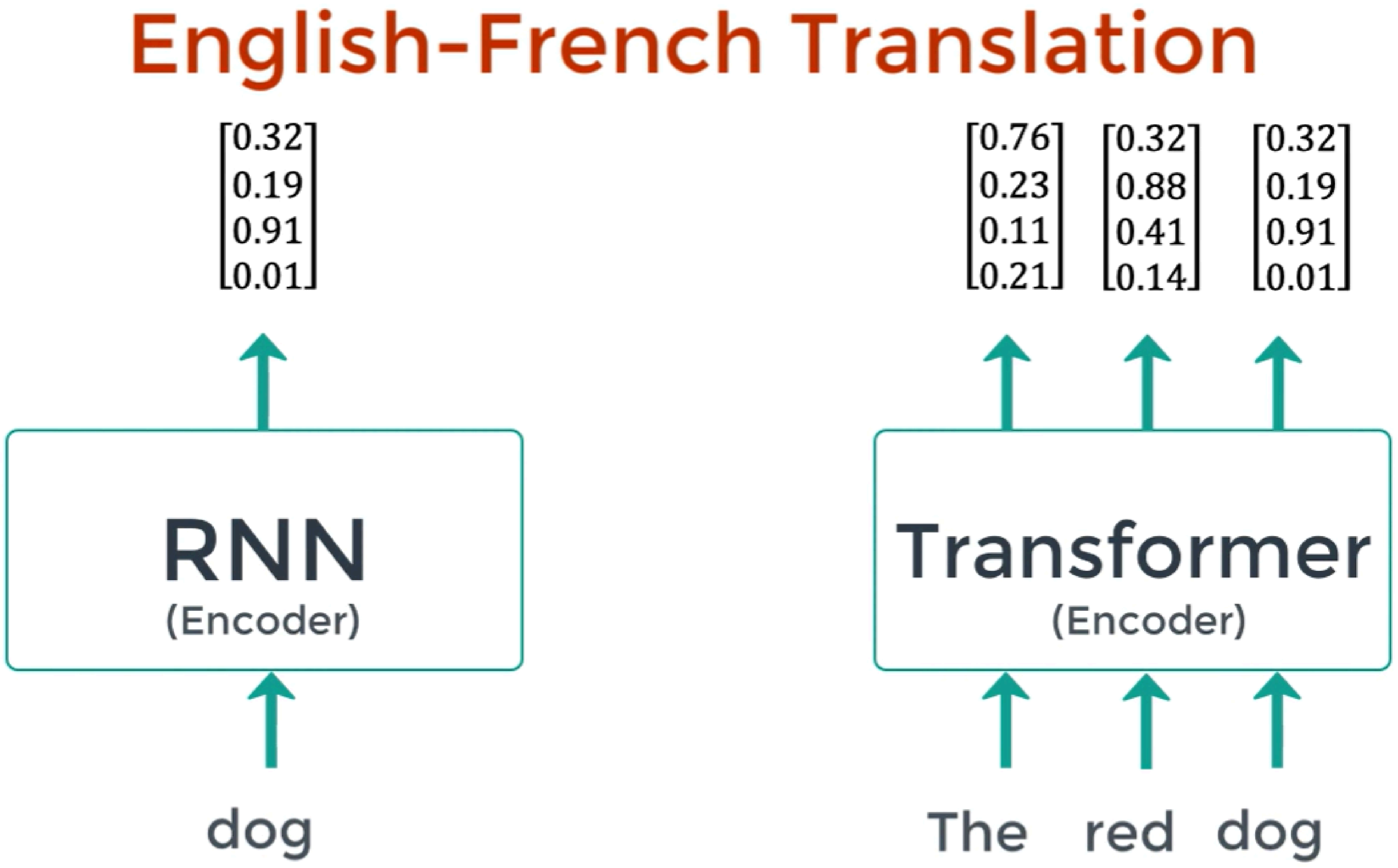
- Enables **high parallelization** by removing sequential operations, allowing significantly faster training compared to RNN models.
- No recurrence or convolution, only **self attention and feed-forward layers.**

Diagrams: Vaswani et al., 2023, Attention Is All You Need, Cornell University Library, arXiv.org, Ithaca.

THE FUNDAMENTAL SHIFT

- RNN: Process tokens **sequentially** one at a time. Context builds over time, but long-range dependencies get harder to capture.

- Transformer: Utilizes **self-attention**. Each token attends to every other simultaneously. **Context is built in parallel.** The model learns what to focus on, regardless of position.



Diagrams: CodeEmporium (2020) Transformer Neural Networks - Explained! (Attention is all you need). https://www.youtube.com/watch?v=TQQlZhbC5ps.

WHY SELF-ATTENTION?

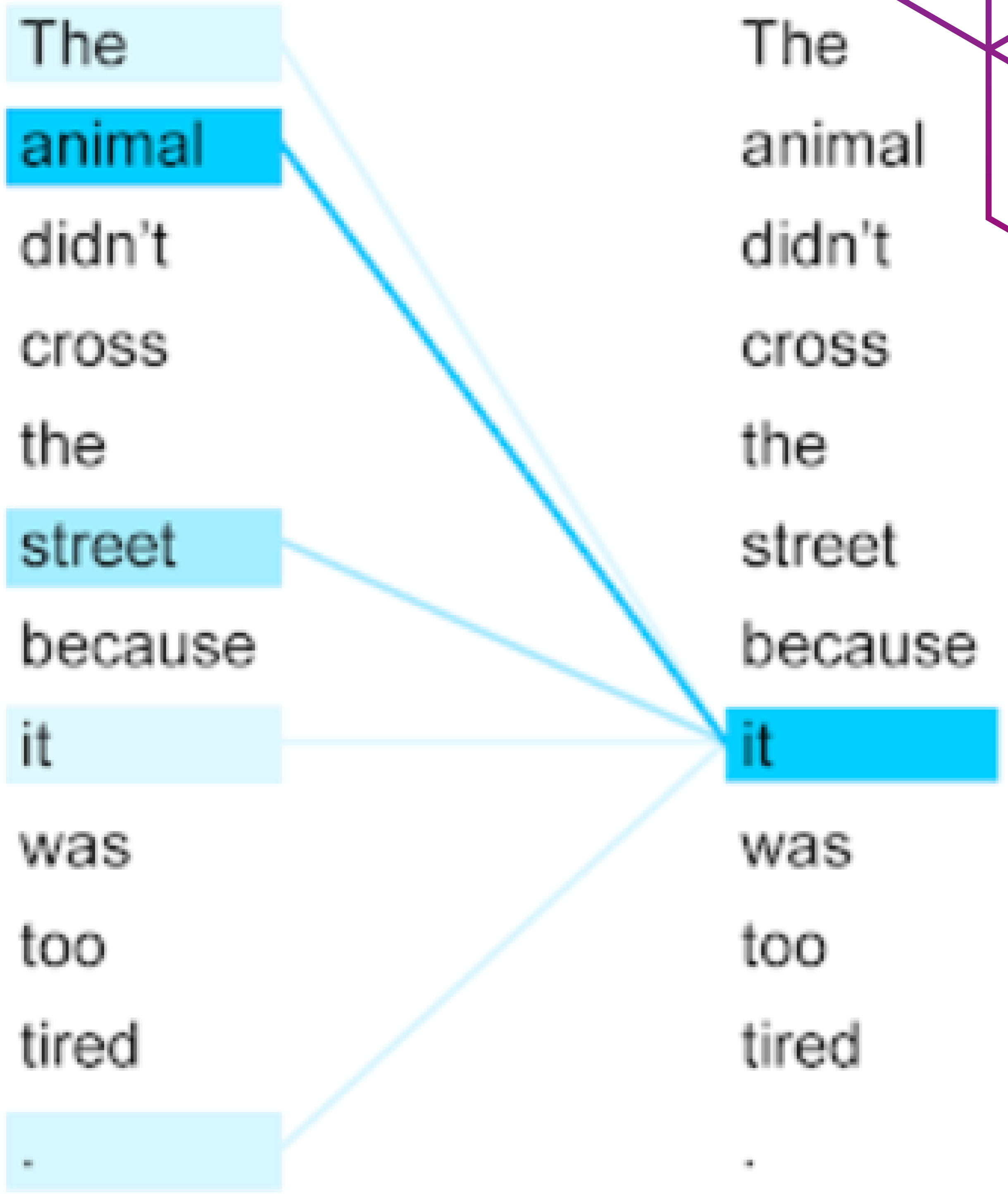
"Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence." (Vaswani et al., 2017)

Consider the following sentence as an input we want to translate:

"THE ANIMAL DIDN'T CROSS THE STREET BECAUSE IT WAS TIRED"

What does the "**IT**" in this sentence refer to?

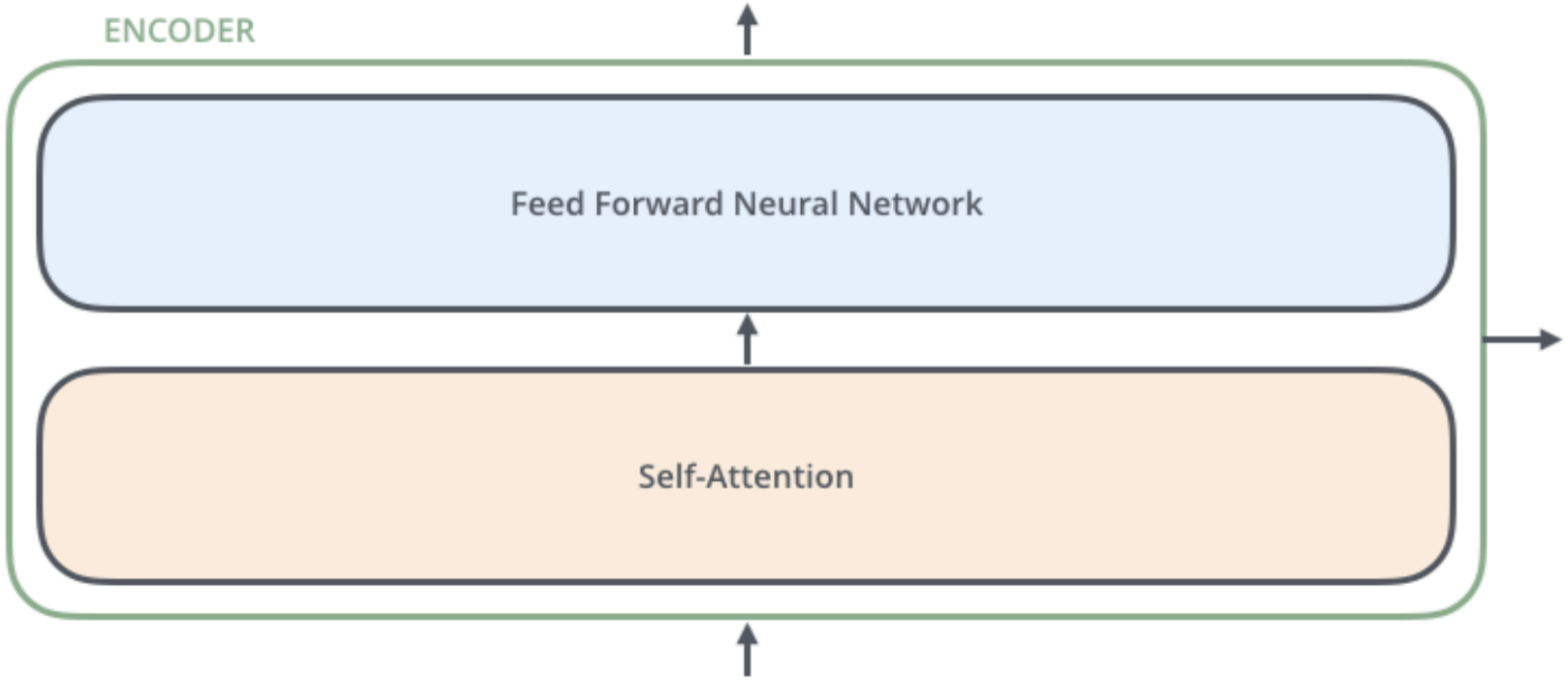
When the model processes the word "**IT**", self-attention allows for it to associate the word "**IT**" with "**ANIMAL**".

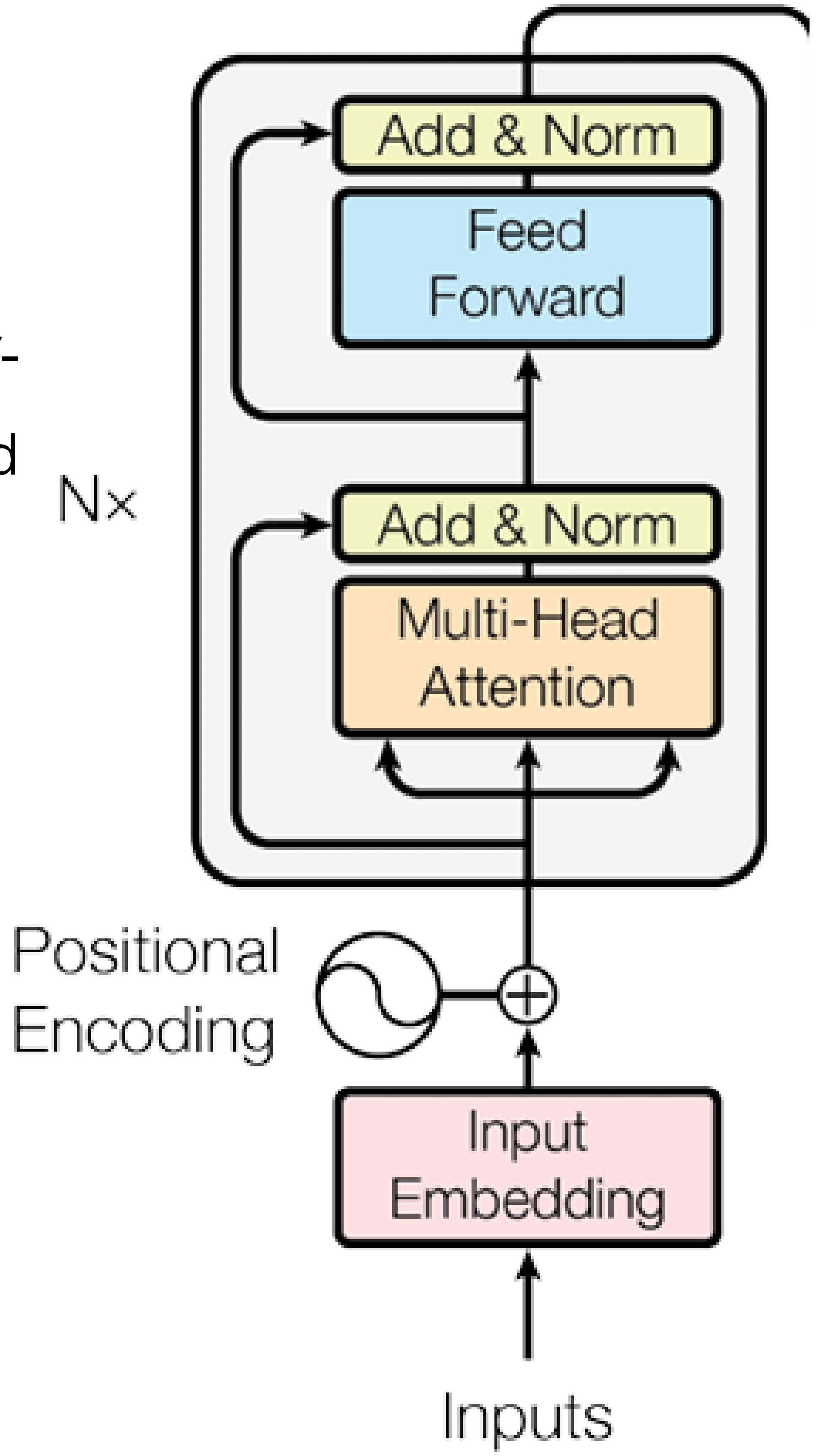


Diagrams: Uszkoreit, 2017, Google Research Blog, Transformer: A Novel Neural Network Architecture for Language Understanding, Available at: https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/

ENCODER

- Encoder is made up of stack of N=6 identical layers.
- In each layer there are two sub-layers which include multi-head self-attention mechanism and a simple position-wise connected feed forward network.

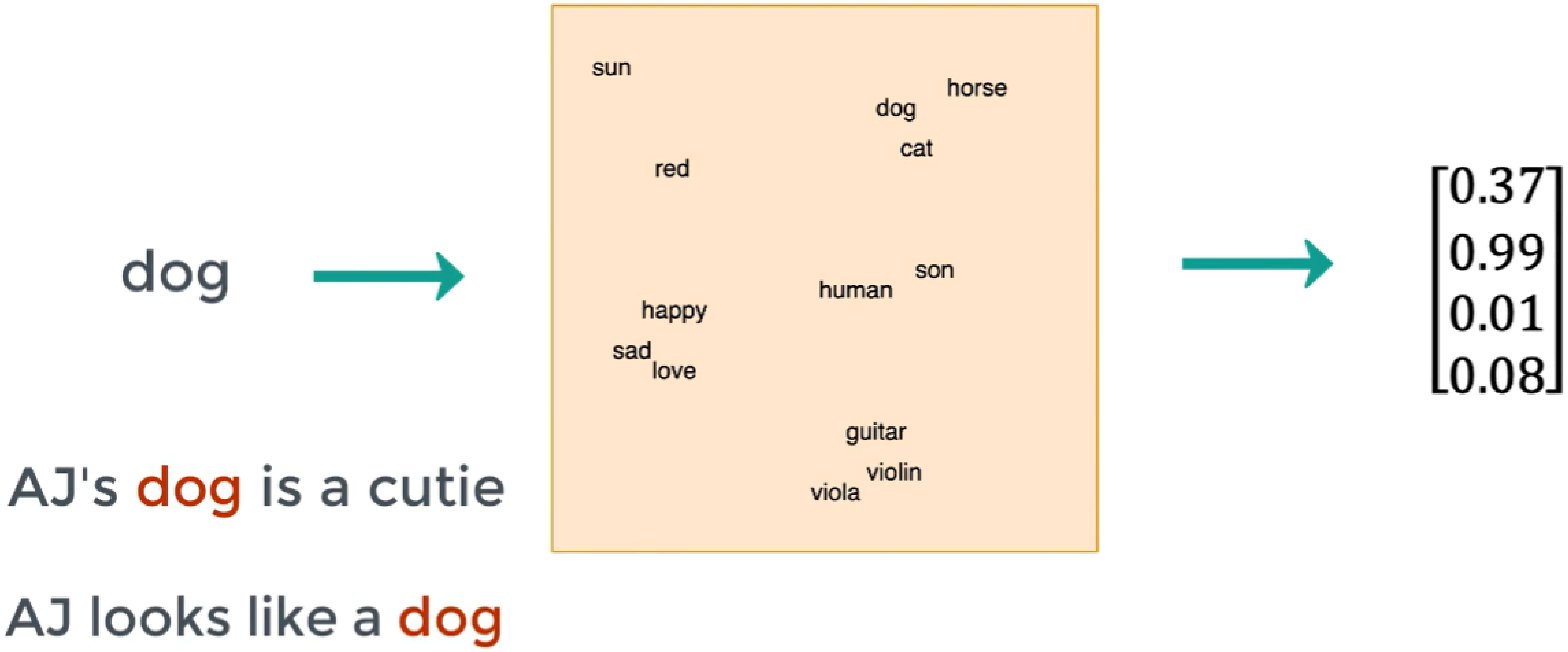




Diagrams: Vaswani et al., 2023, Attention Is All You Need, Cornell University Library, arXiv.org, Ithaca.
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

INPUT EMBEDDINGS

- In NLP applications, words are represented as numeric vectors which allows similar words to have vector representations that are similar.
- We begin by converting each input word into a vector, this embedding happens only once in the bottom-most encoder.
- These embeddings are learned during training and capture semantic information.



$$\begin{bmatrix}0.37\\0.99\\0.01\\0.08\end{bmatrix}$$

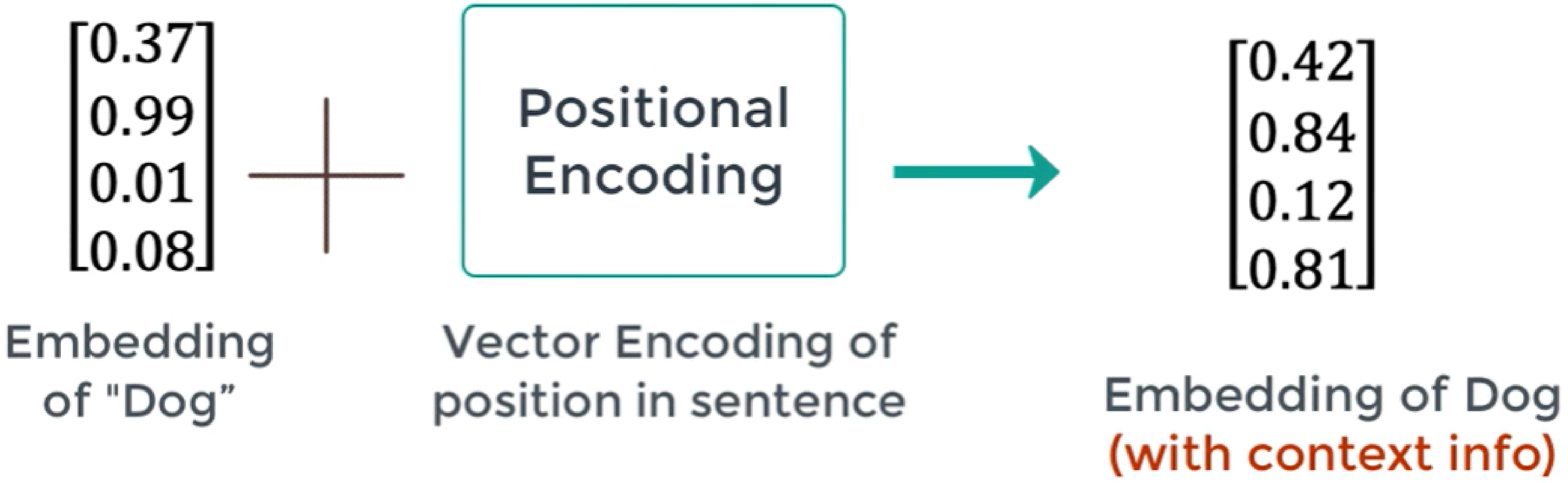
AJ's dog is a cutie

AJ looks like a dog

Diagrams: CodeEmporium (2020) Transformer Neural Networks - Explained! (Attention is all you need). https://www.youtube.com/watch?v=TQQlZhbC5ps.

POSITION ENCODING

- Since the model contains no recurrence or convolution, to **account for the order of the words** in the input sequence, the transformer **adds vector to each input embedding**.
- The model learns the pattern from these vectors that helps it determine each word’s position or the distance between the words in a sequence.
- In this paper, they used **sine and cosine functions of different frequencies** where pos is the position and i is the dimension.



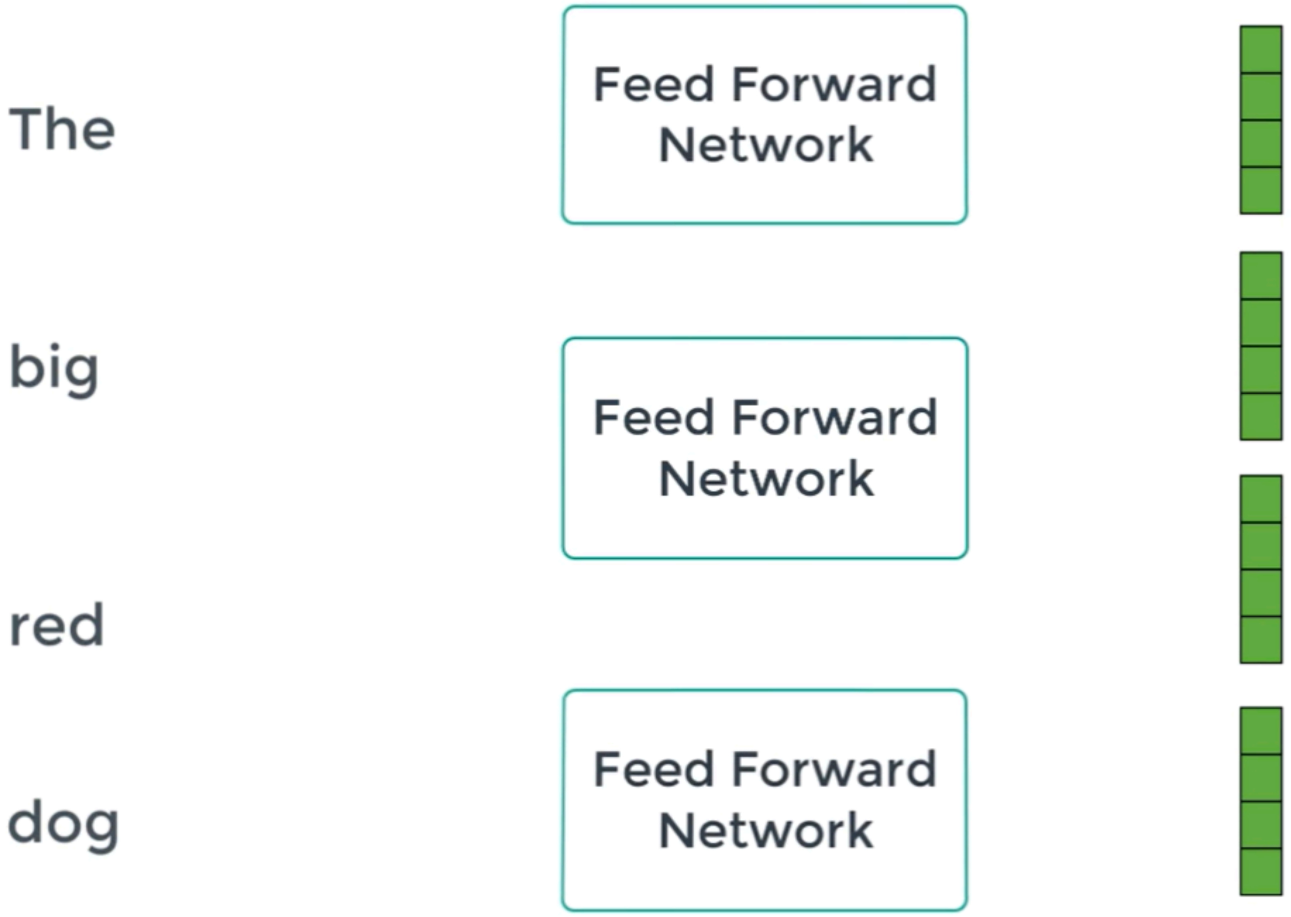
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Diagrams: CodeEmporium (2020) Transformer Neural Networks - Explained! (Attention is all you need). https://www.youtube.com/watch?v=TQQlZhbC5ps.
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

FEED FORWARD NETWORKS

- While attention layer enables each token to incorporate context from entire sequence, feed-forward network is responsible for non-linear transformation and abstraction at the level of each token independent of it’s neighbors.
- This network is applied identically and independently to each position in the sequence.
- Consist of two linear transformation with non-linear activation (ReLU) in between.

FFN(x) = max(0xW1 +b1)W2 +b2



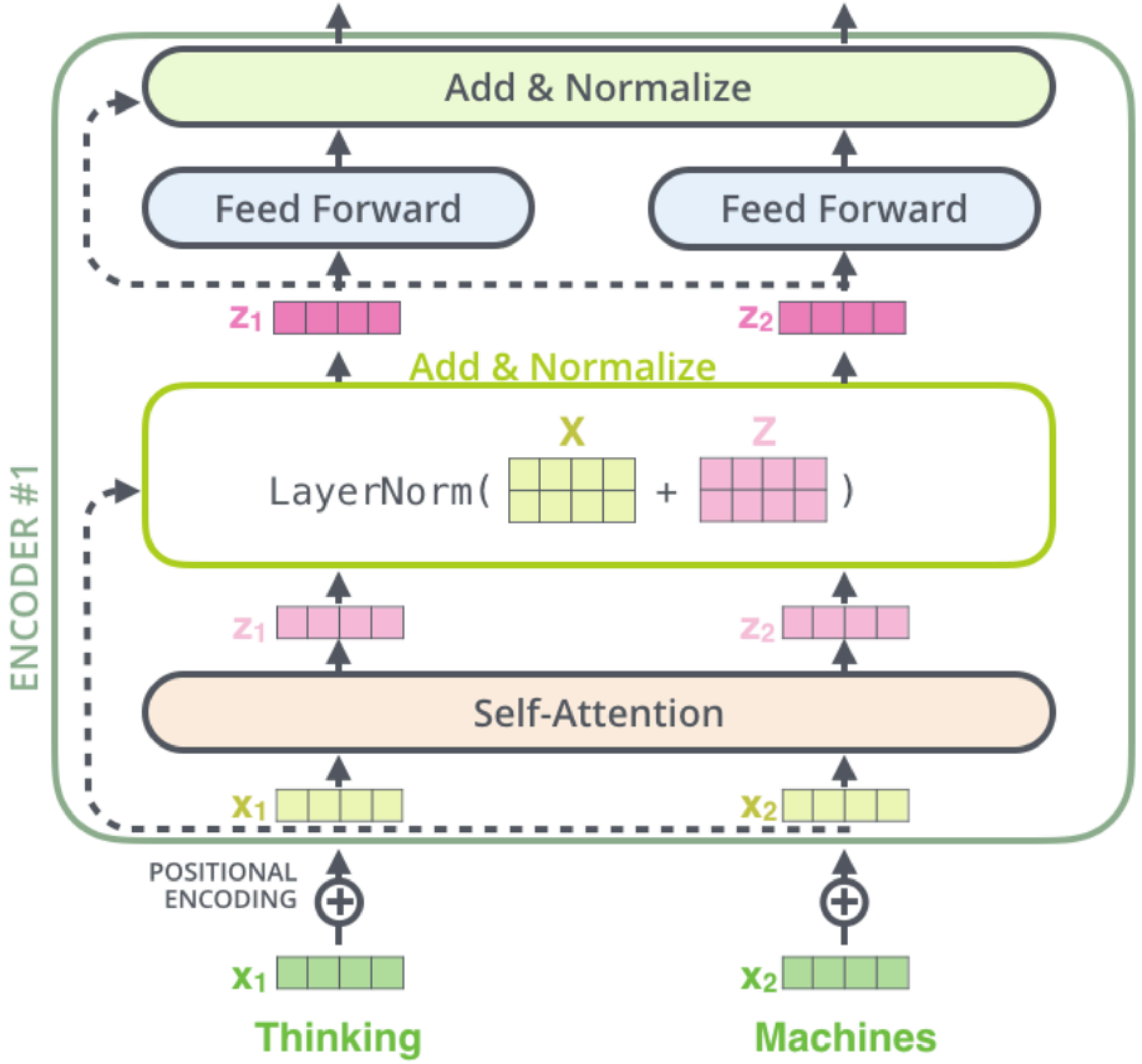
Diagrams: CodeEmporium (2020) Transformer Neural Networks - Explained! (Attention is all you need). https://www.youtube.com/watch?v=TQQlZhbC5ps.

ADD & NORM

In each layer, the output of every sub-layer is passed through Add & Norm → Residual addition followed by Layer normalization.

LayerNorm(x + Sublayer(x))

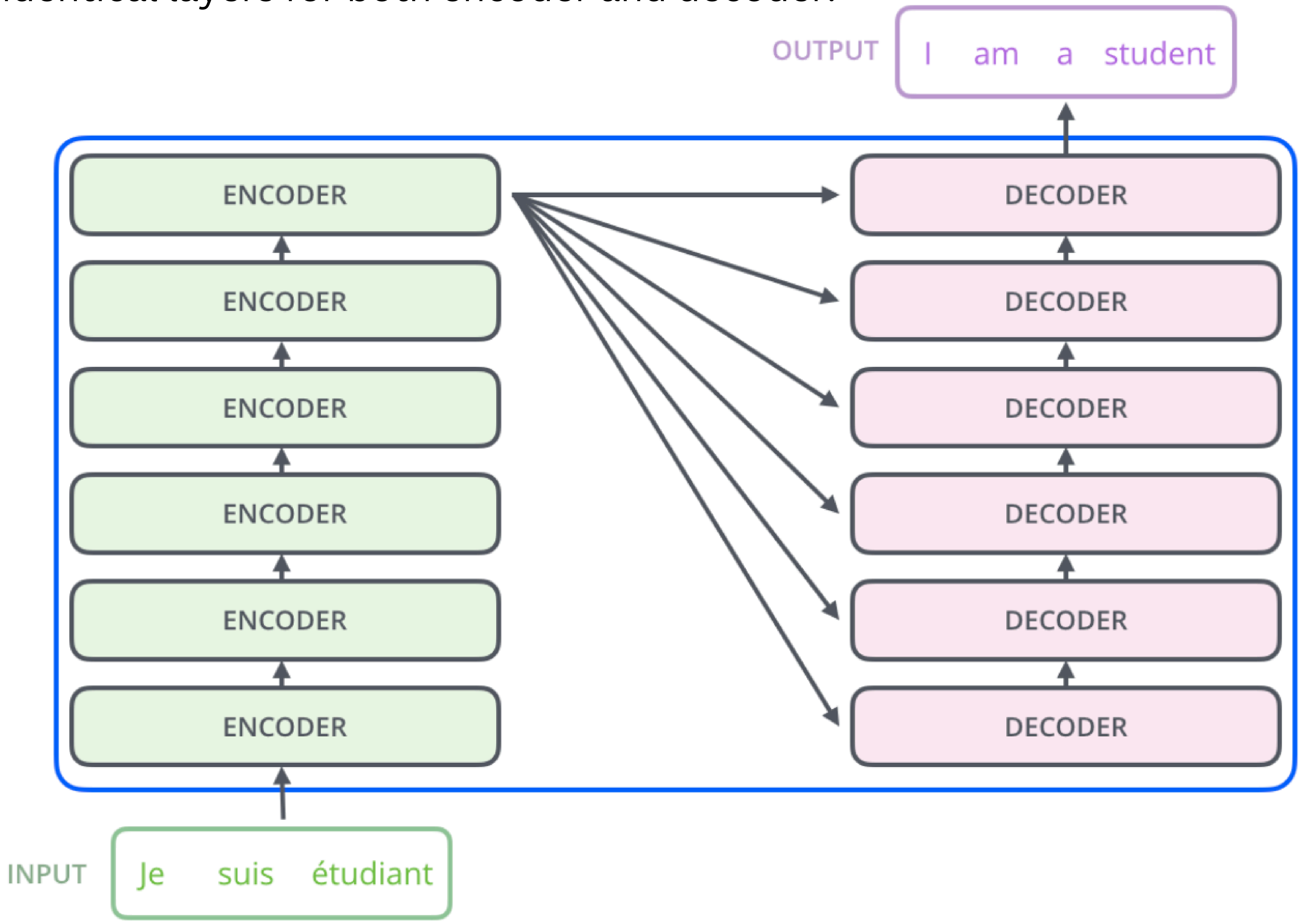
- The input to the sub-layer [x] is passed through transformation. The output of that transformation [Sublayer (x)] is added back to original input [x].
- The resulting vector is then passed through Layer Normalization.
- Helps ensure new information is integrated while preserving prior knowledge.



Diagrams: Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

STACKING LAYERS (Nx)

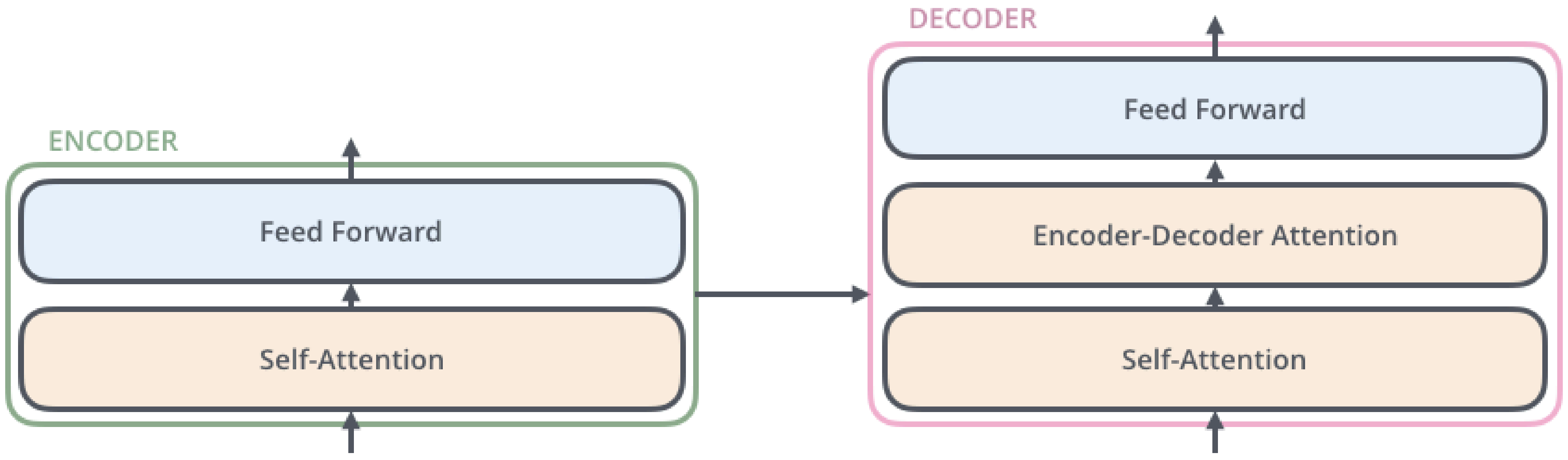
- The transformer utilizes a stack of layers depicted as Nx where N is the number of times the encoder and decoder are stacked on top of each other.
- The paper uses N=6 identical layers for both encoder and decoder.

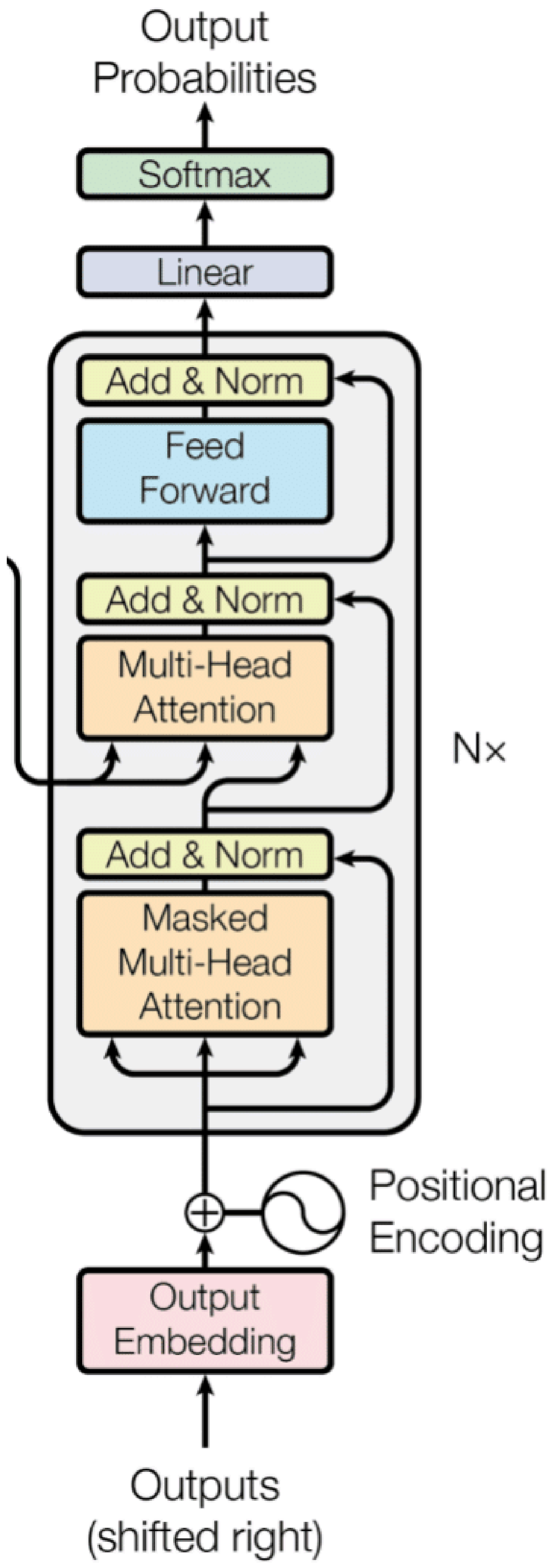


Diagrams: Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

DECODER

- Each decoder layer consists of masked self-attention, encoder-decoder attention, and a feed-forward network, enabling it to generate outputs autoregressively while attending to encoder outputs.
- Masked self-attention ensures the decoder only attends to earlier positions in the output sequence, preserving the autoregressive property during generation.

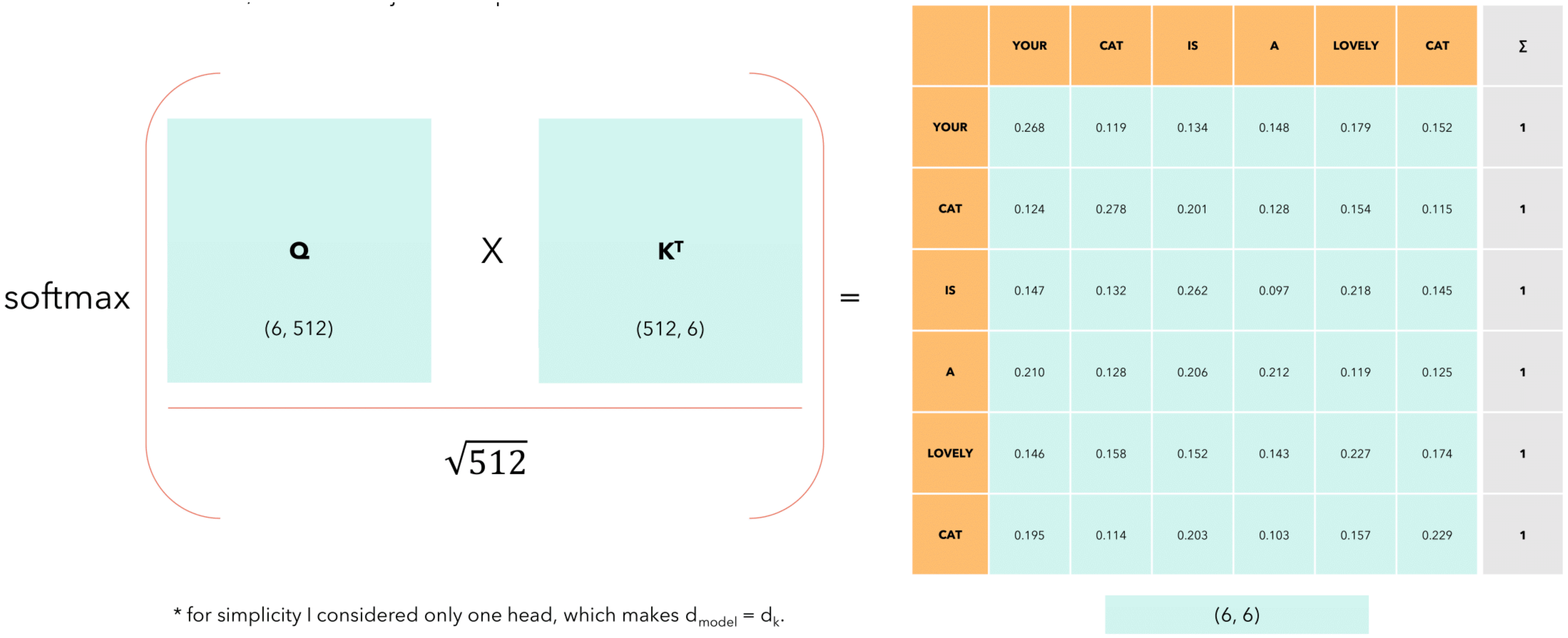




Diagrams: Vaswani et al., 2023, Attention Is All You Need, Cornell University Library, arXiv.org, Ithaca.
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

SELF ATTENTION

- Self Attention allows the model to relate words to each other.
- In this case we consider Seq=6 and dmodel=512.
- The matrices Q, K and V are input sentence.



$$\text{softmax}\left(\frac{\underset{(6,\,512)}{Q} \;\text{X}\; \underset{(512,\,6)}{K^T}}{\sqrt{512}}\right) =$$

	YOUR	CAT	IS	A	LOVELY	CAT	Σ
YOUR	0.268	0.119	0.134	0.148	0.179	0.152	1
CAT	0.124	0.278	0.201	0.128	0.154	0.115	1
IS	0.147	0.132	0.262	0.097	0.218	0.145	1
A	0.210	0.128	0.206	0.212	0.119	0.125	1
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174	1
CAT	0.195	0.114	0.203	0.103	0.157	0.229	1

(6, 6)

* for simplicity I considered only one head, which makes $d_{model} = d_k$.

Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

SELF ATTENTION

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

	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	0.119	0.134	0.148	0.179	0.152
CAT	0.124	0.278	0.201	0.128	0.154	0.115
IS	0.147	0.132	0.262	0.097	0.218	0.145
A	0.210	0.128	0.206	0.212	0.119	0.125
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174
CAT	0.195	0.114	0.203	0.103	0.157	0.229

(6, 6)

X

V

(6, 512)

=

Attention

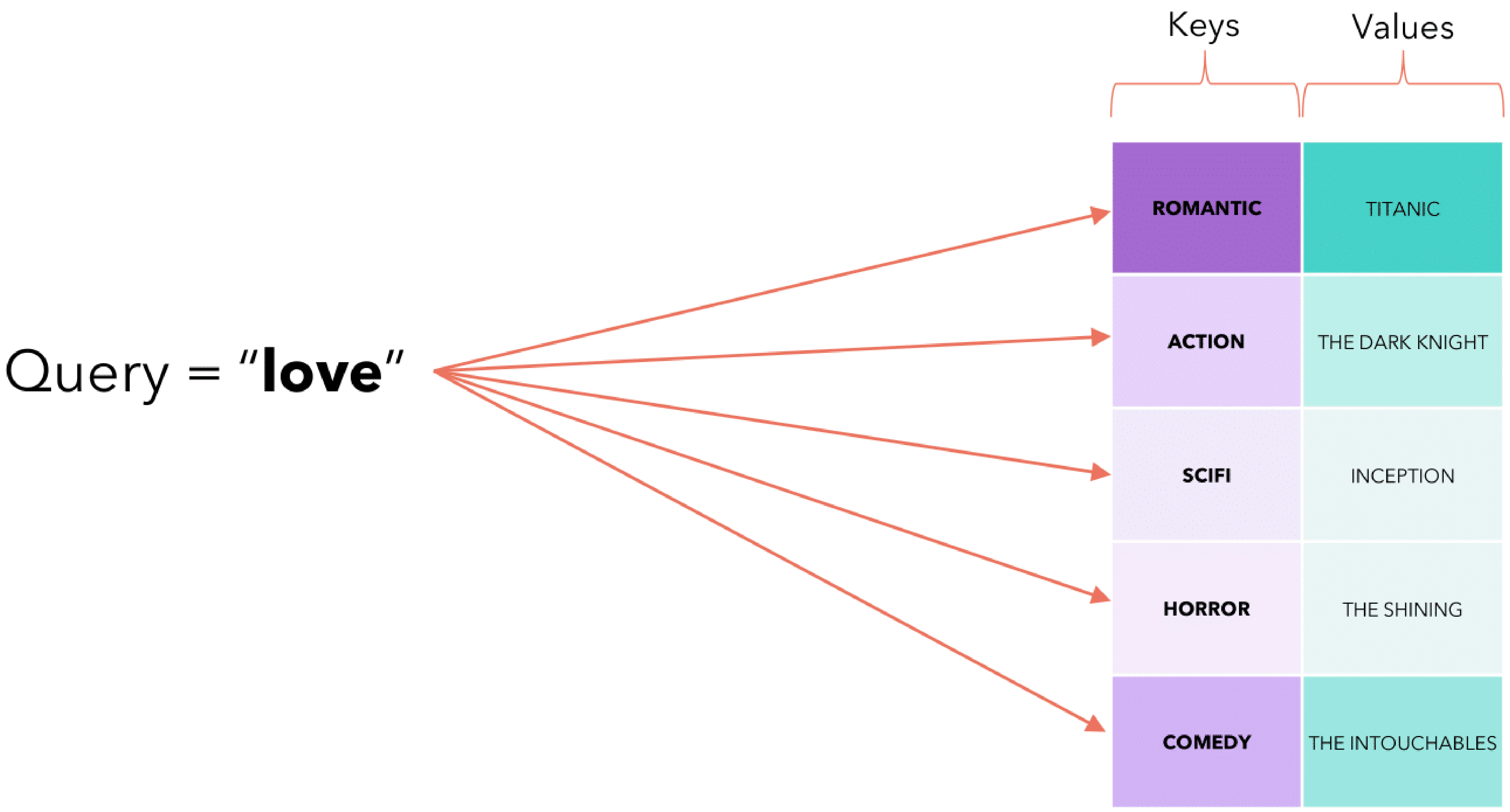
(6, 512)

Each row in this matrix captures not only the meaning (given by the embedding) or the position in the sentence (represented by the positional encodings) but also each word's interaction with other words.

Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

QUERY, KEYS AND VALUES

- In attention mechanisms, each input is projected into three vectors: a Query, a Key , and a Value.
- Attention is computed by comparing the Query with all Keys to produce weights, which are then used to sum the corresponding Values.
- This allows the model to dynamically focus on relevant parts of the sequence when processing each token.



Query = "**love**"

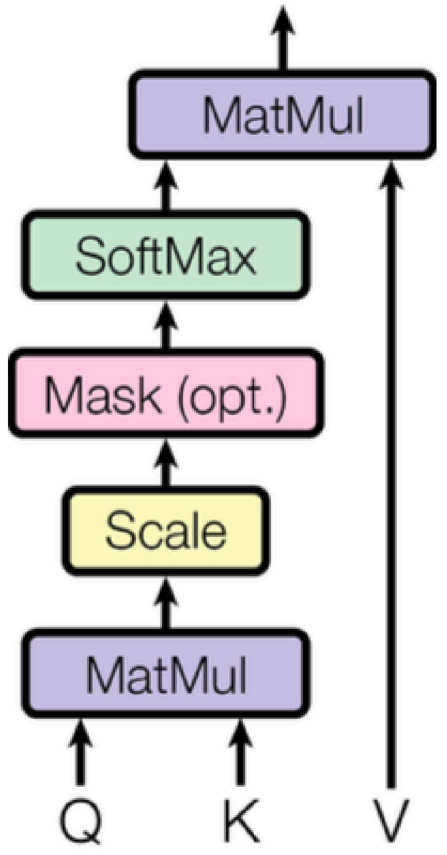
Keys	Values
ROMANTIC	TITANIC
ACTION	THE DARK KNIGHT
SCIFI	INCEPTION
HORROR	THE SHINING
COMEDY	THE INTOUCHABLES

Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

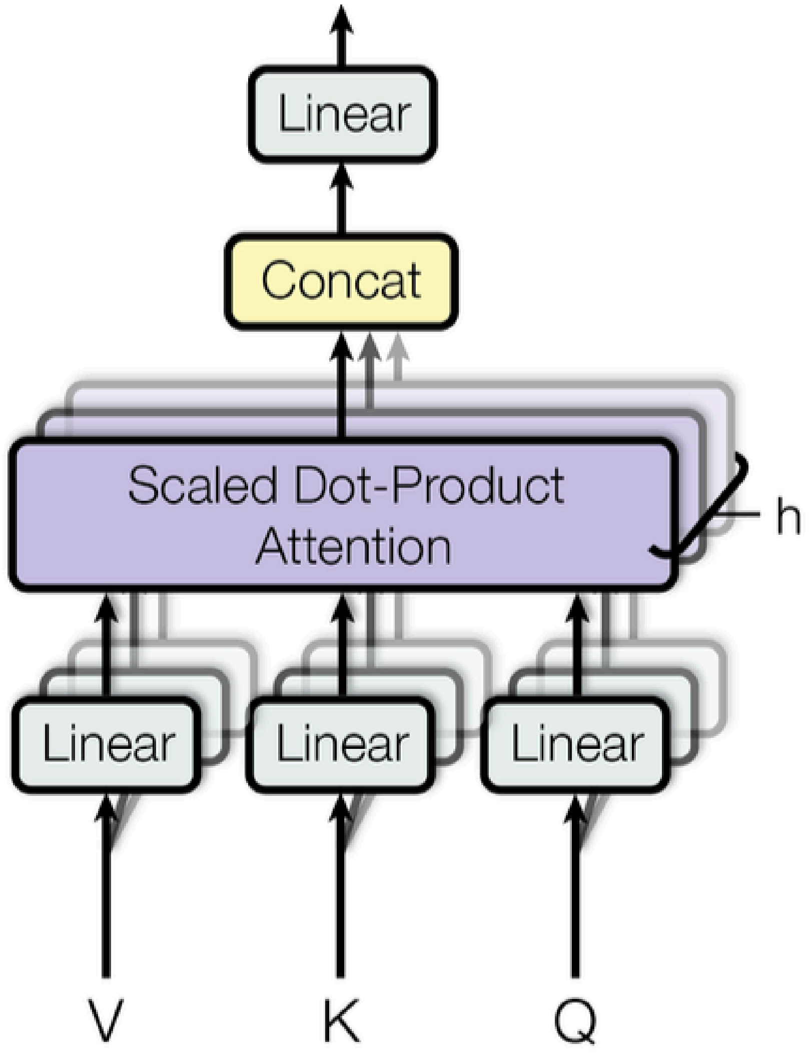
MULTI-HEAD ATTENTION

- Multi-head attention allows the model to focus on different positions of the input sequence simultaneously, capturing various types of relationships.
- This enriches the representation by allowing each head to attend to different subspaces of the input.

Scaled Dot-Product Attention

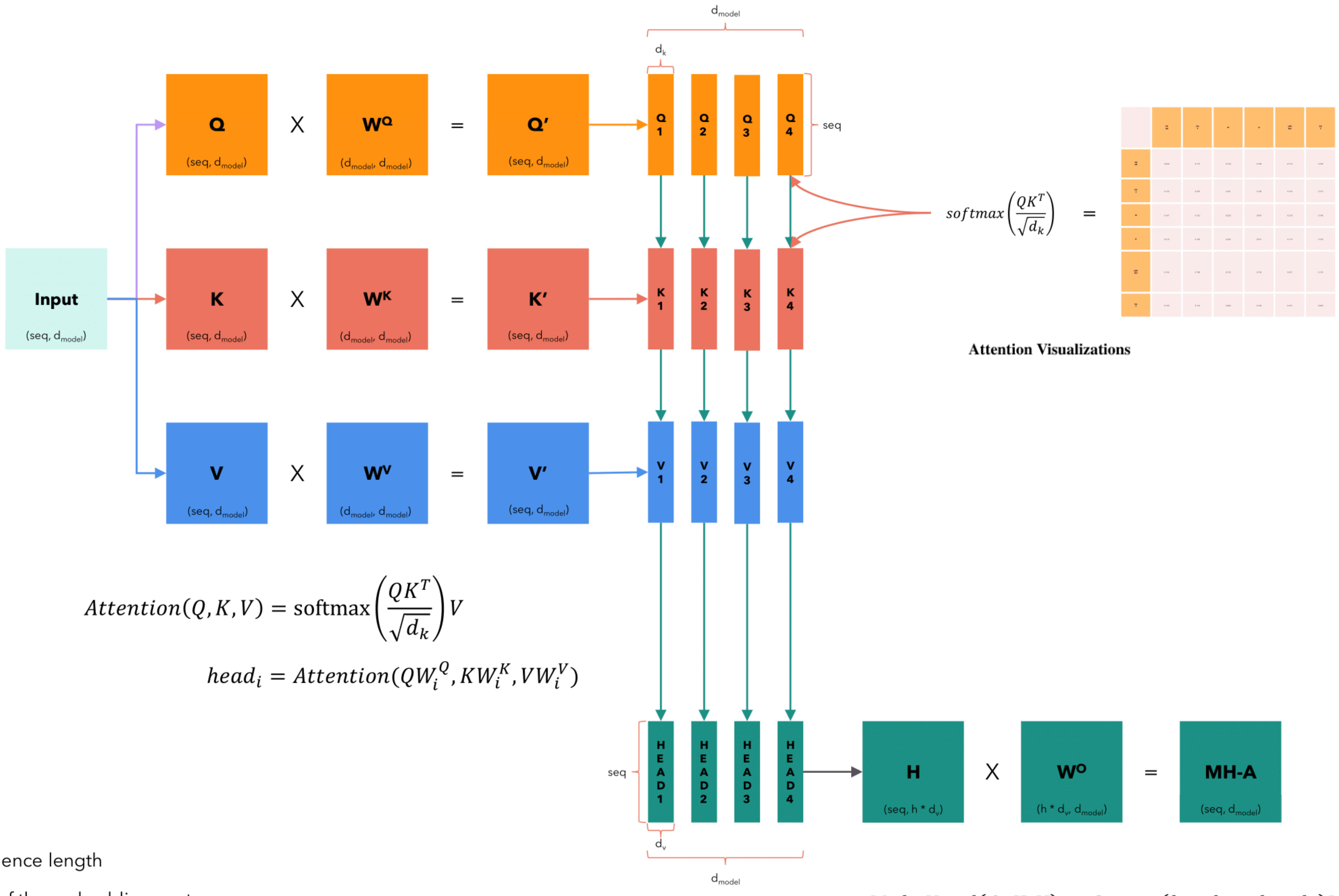


Multi-Head Attention



Diagrams: Vaswani et al., 2023, Attention Is All You Need, Cornell University Library, arXiv.org, Ithaca.

MULTI-HEAD ATTENTION



Attention Visualizations

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

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

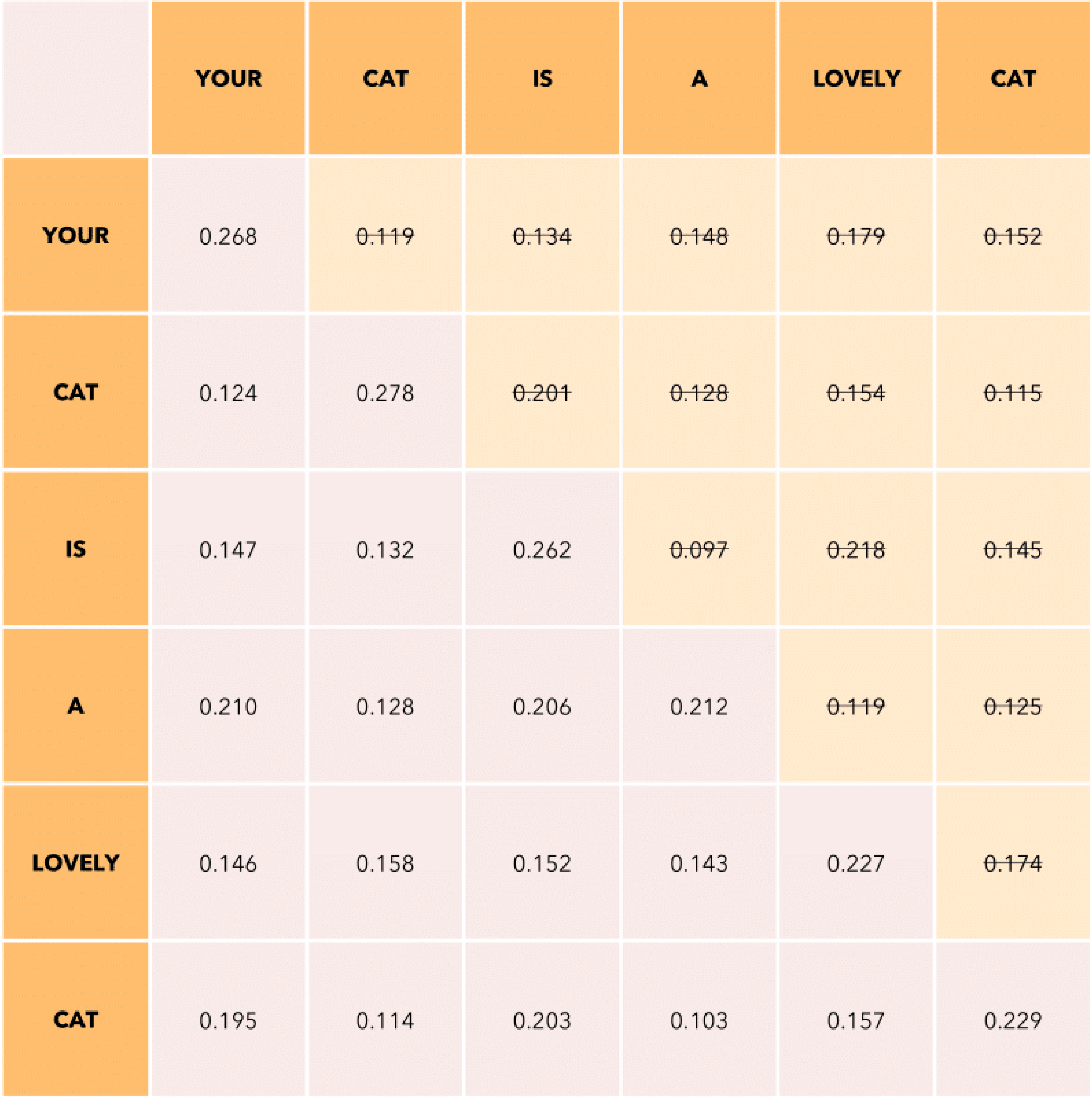
$$MultiHead(Q,K,V) = Concat(head_1\ \ldots head_h)W^O$$

seq	= sequence length
d_{model}	= size of the embedding vector
h	= number of heads
$d_{k = }d_v$	= d_{model} / h

Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

MASKED MULTI-HEAD ATTENTION

- Masked multi-head attention prevents the decoder from attending to future tokens by applying a mask over the attention weights.
- Each head still learns different attention patterns, but only within the allowed (past) context.



	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	~~0.119~~	~~0.134~~	~~0.148~~	~~0.179~~	~~0.152~~
CAT	0.124	0.278	~~0.201~~	~~0.128~~	~~0.154~~	~~0.115~~
IS	0.147	0.132	0.262	~~0.097~~	~~0.218~~	~~0.145~~
A	0.210	0.128	0.206	0.212	~~0.119~~	~~0.125~~
LOVELY	0.146	0.158	0.152	0.143	0.227	~~0.174~~
CAT	0.195	0.114	0.203	0.103	0.157	0.229

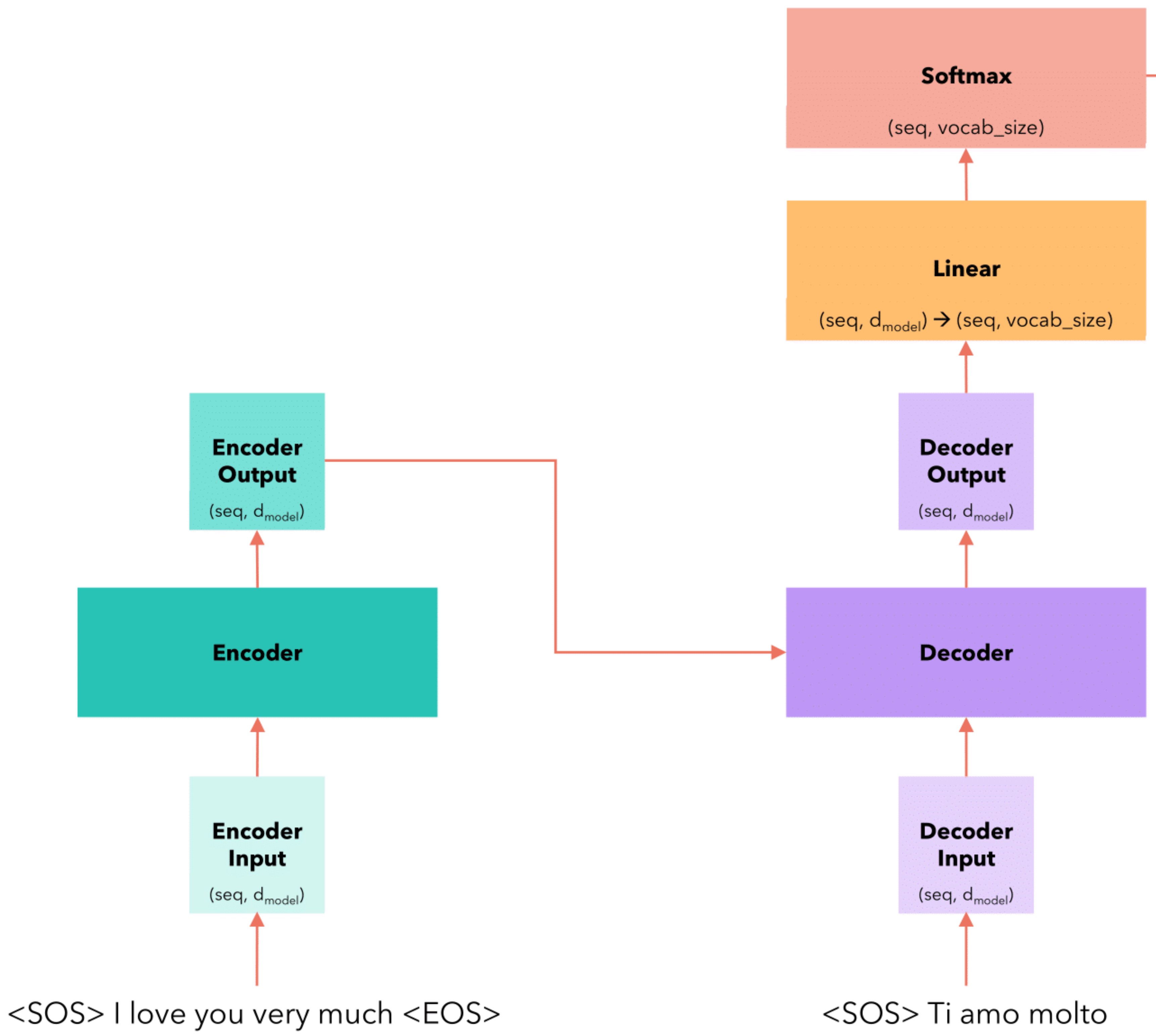
Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

TRAINING

Ti amo molto <EOS>

* This is called the "label" or the "target"

Cross Entropy Loss



Softmax

(seq, vocab_size)

Linear

(seq, d_{model}) → (seq, vocab_size)

Encoder Output

(seq, d_{model})

Decoder Output

(seq, d_{model})

Encoder

Decoder

Encoder Input

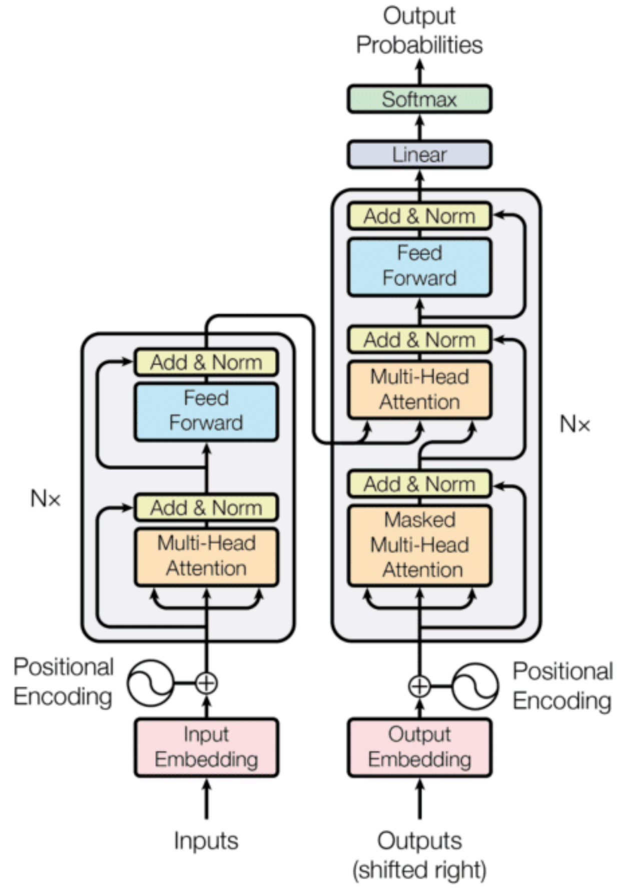
(seq, d_{model})

Decoder Input

(seq, d_{model})

<SOS> I love you very much <EOS>

<SOS> Ti amo molto



We prepend the <SOS> token at the beginning. That's why the paper says that the decoder input is shifted right.

Diagram: Umar Jamil (2023) Attention is all you need (Transformer) - Model explanation (including math), Inference and Training. https://www.youtube.com/watch?v=bCz4OMemCcA.

TRAINING

- After passing through the final decoder layer, output vectors are sent through a linear (fully connected) layer that maps them to the vocabulary size.
- A softmax function is then applied to convert these logits into probabilities over the vocabulary.
- During training, the model uses teacher forcing and is optimized using cross-entropy loss between predicted and target tokens.

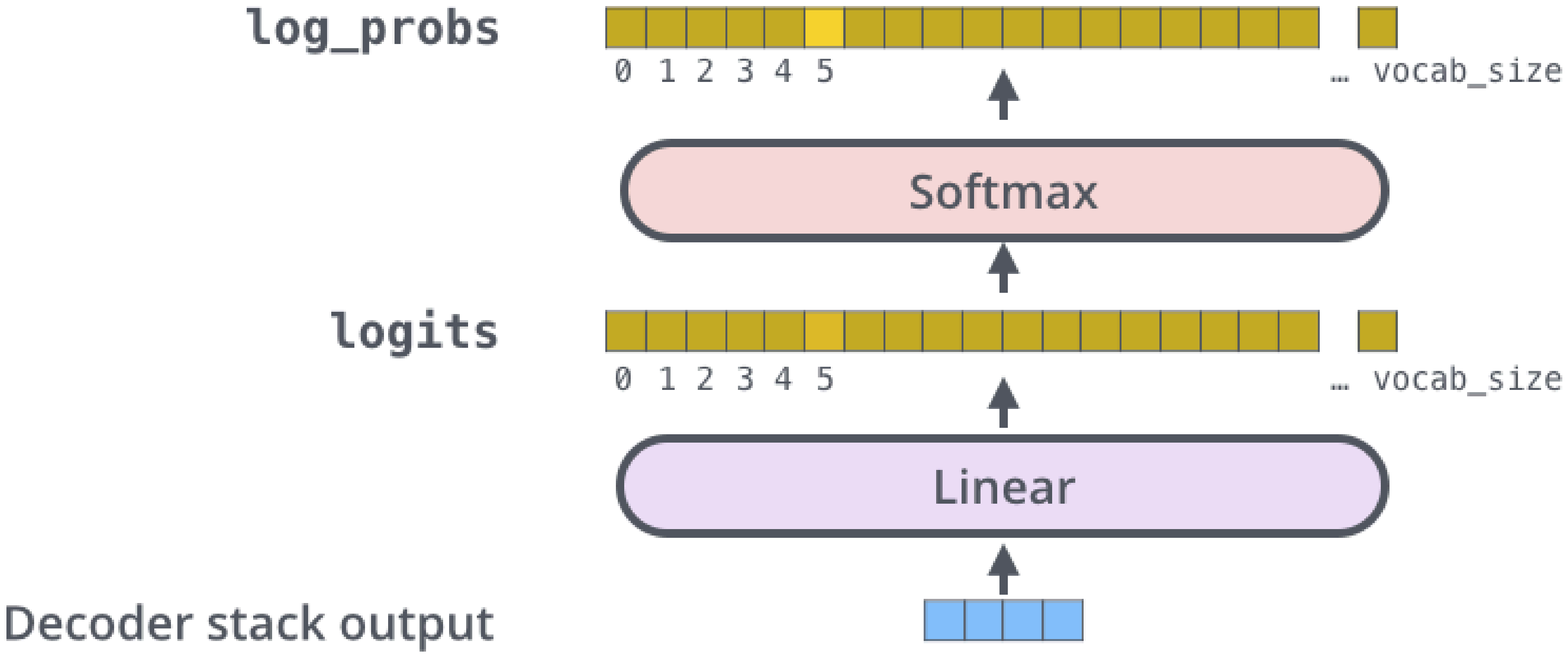


Diagram: Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

RESULTS

- The Transformer (big) model achieved a BLEU score of 28.4 on WMT 2014 English-to-German, outperforming all previous models, including ensembles, by more than 2 BLEU points.
- On the English-to-French task, it reached a BLEU of 41.0, the highest for any single model at the time, and with less than ¼ the training cost of earlier systems.
- Even the base Transformer model outperforms older RNN-based models, while requiring significantly less training time and GPU resources.
- This efficiency comes from removing recurrence and relying solely on attention, enabling much more parallelization during training.

CONCLUSION

- The Transformer introduced a fully attention-based architecture that revolutionized sequence modeling by replacing recurrence and convolution.
- Its superior BLEU scores, faster training, and scalability made it the foundation for nearly all modern language models, including BERT, GPT, T5, and many more.
- This work reshaped the field of NLP, proving that attention alone is not only sufficient, it's more effective.
- Today, the Transformer remains central to progress in machine translation, text generation, code understanding, and even vision and audio tasks.

How do transformers work to outperform traditional NLP models? (2025). https://mariusbrt.com/articles/transformer. Transformers. https://huggingface.co/blog/Esmail-AGumaan/attention-is-all-you-need.