UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

Sesión 3.0

Self-attention

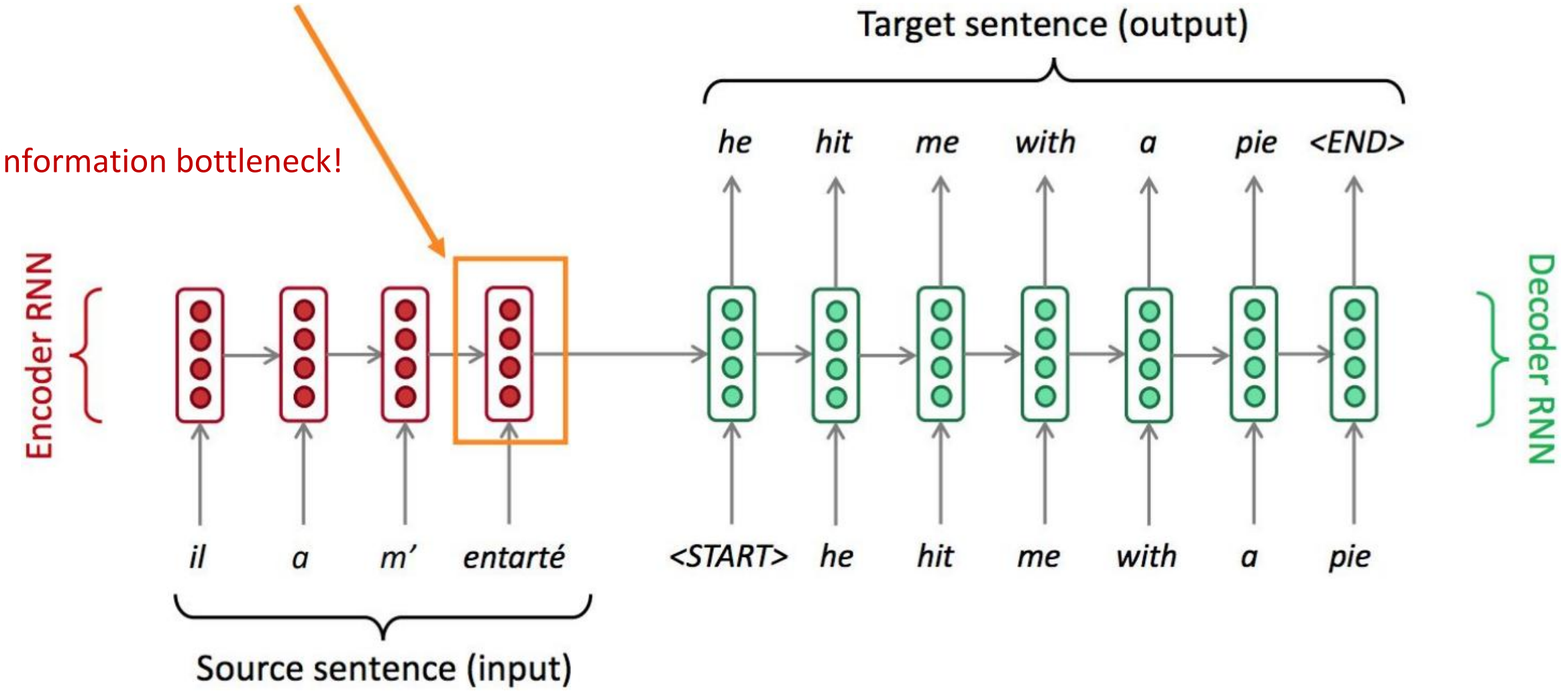
Transformers

TRANSFORMATEC

Sequence-to-*sequence models*

Codificación de la oración fuente.
Debe capturar toda la información
sobre la oración fuente.

Information bottleneck!



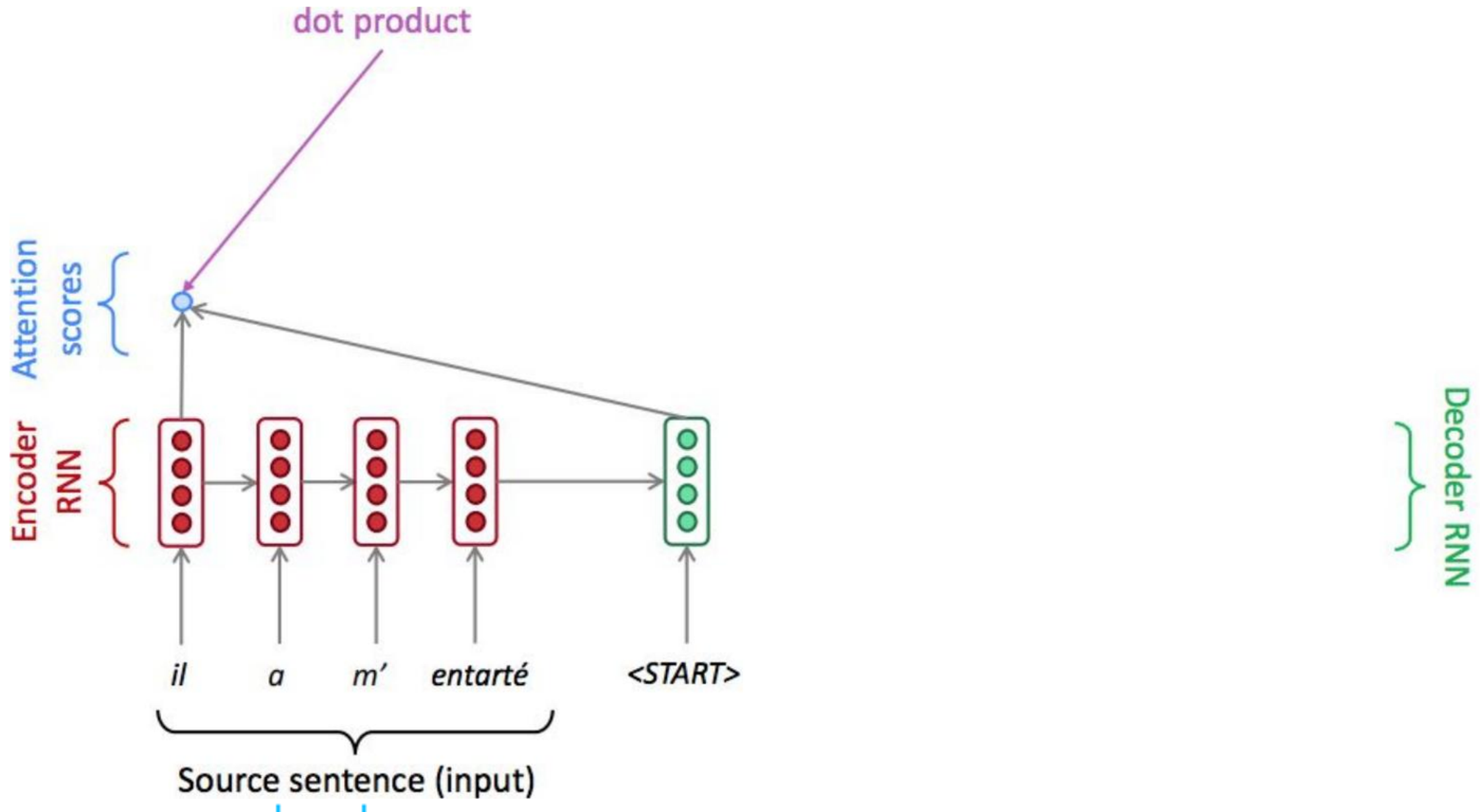
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1.

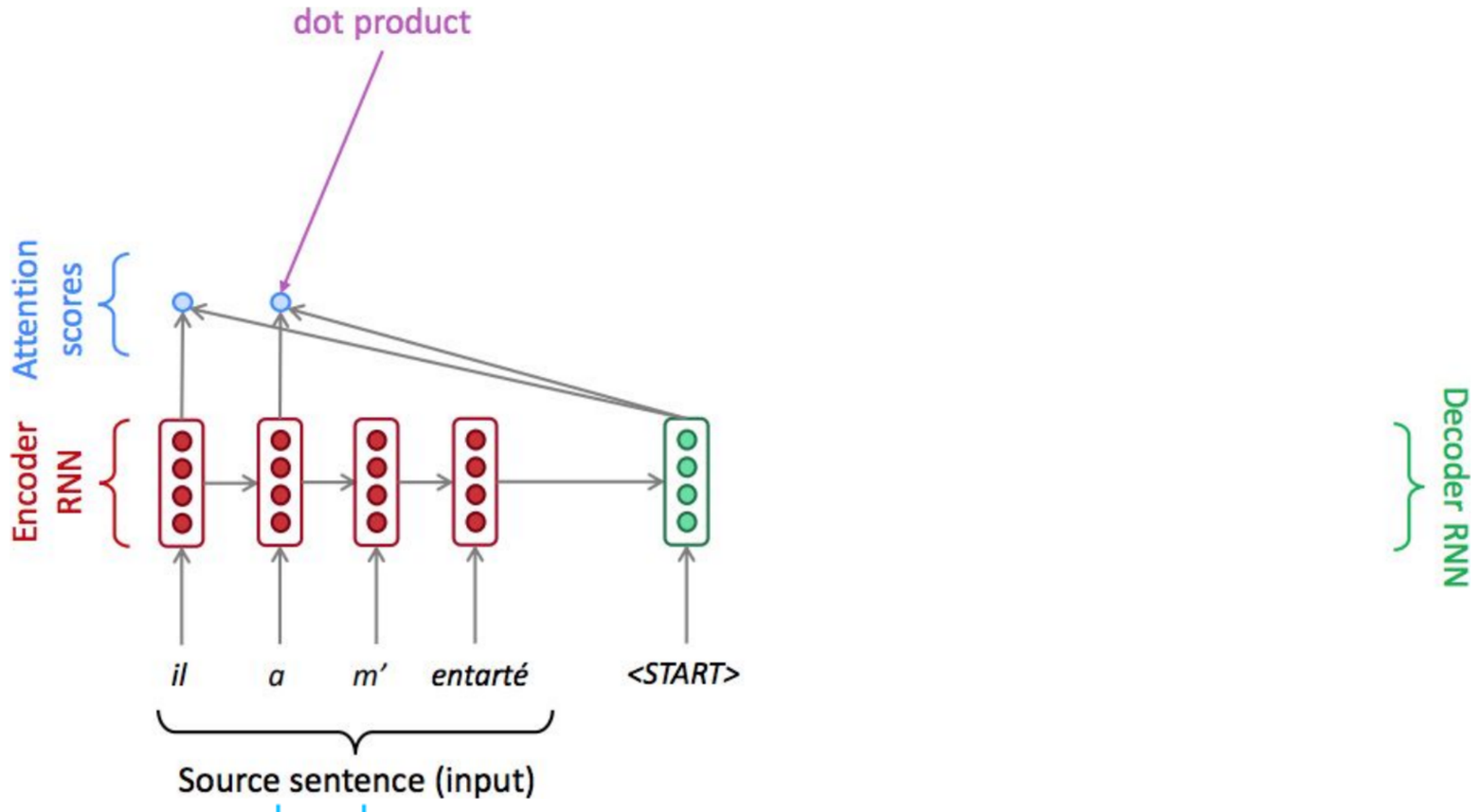
Attention

TRANSFORMATEC

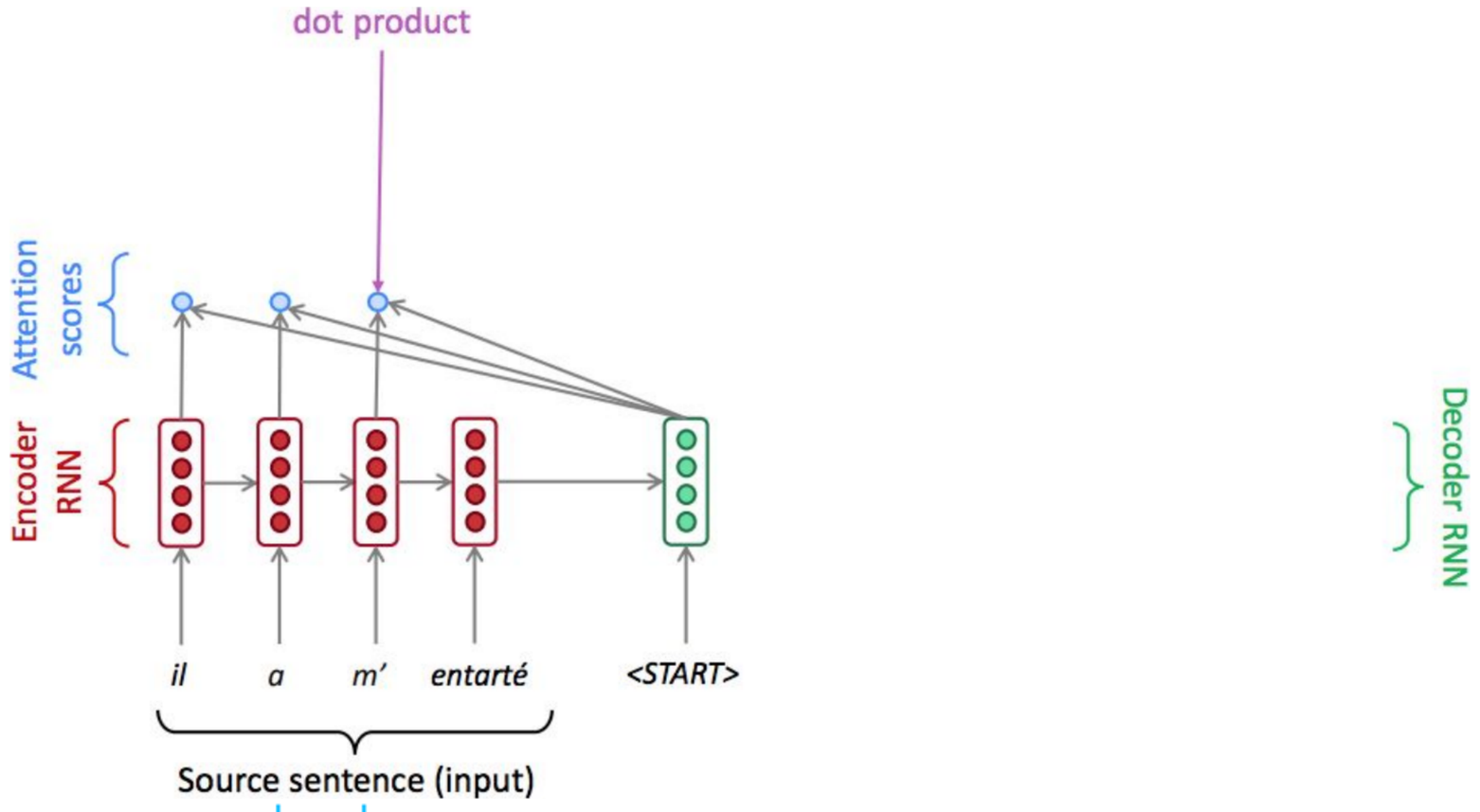
Attention *Mechanism*



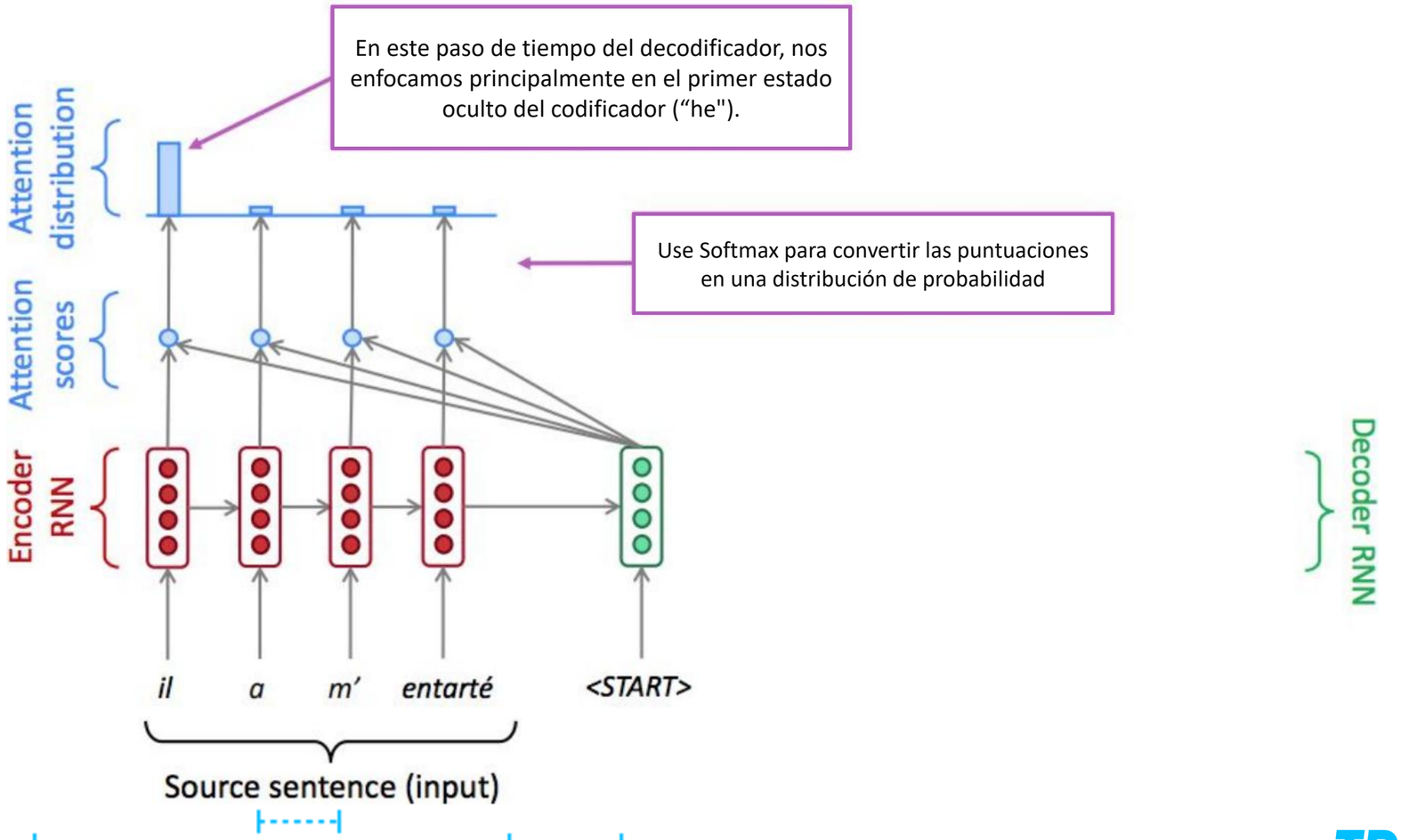
Attention *Mechanism*



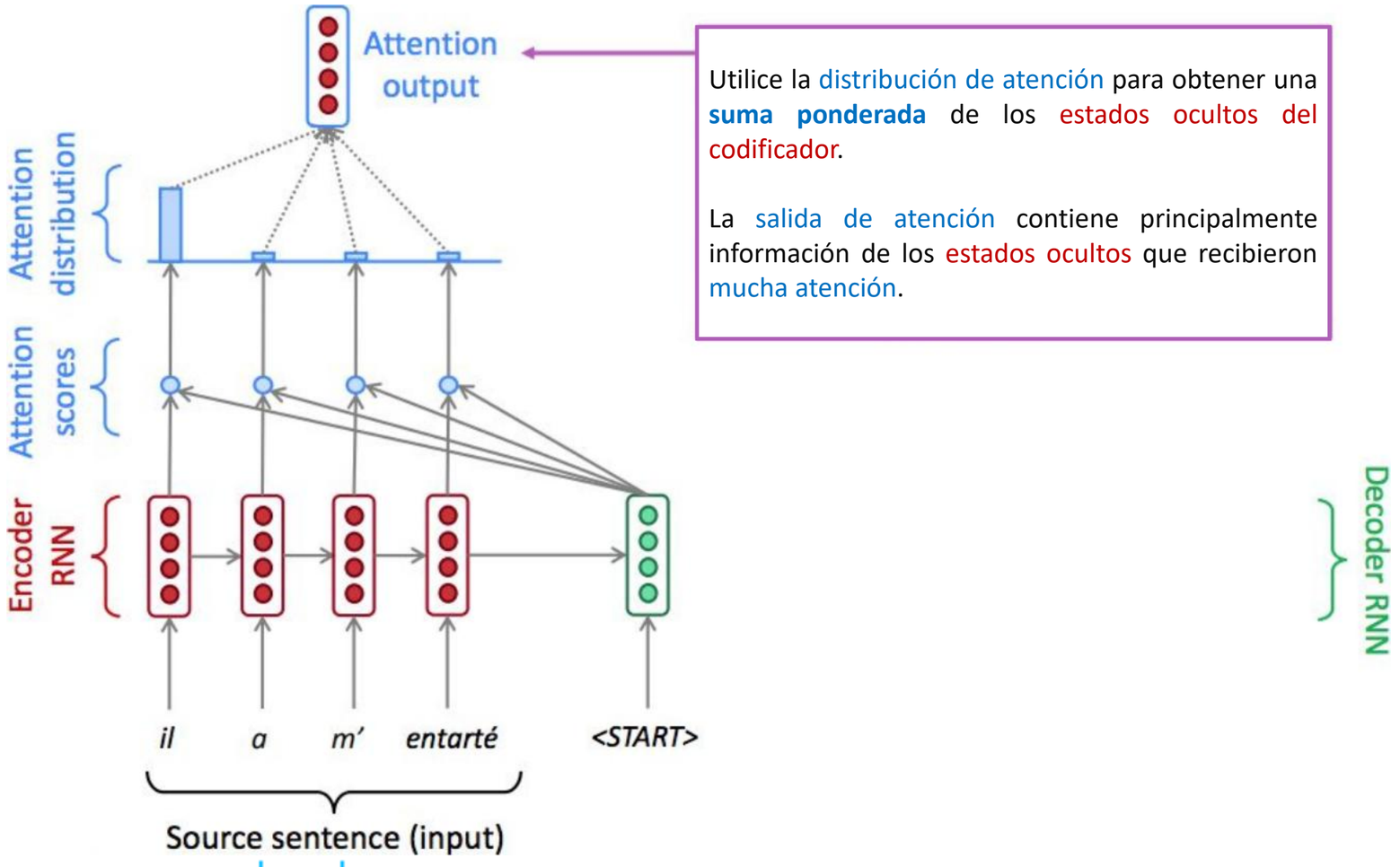
Attention *Mechanism*



Attention *Mechanism*



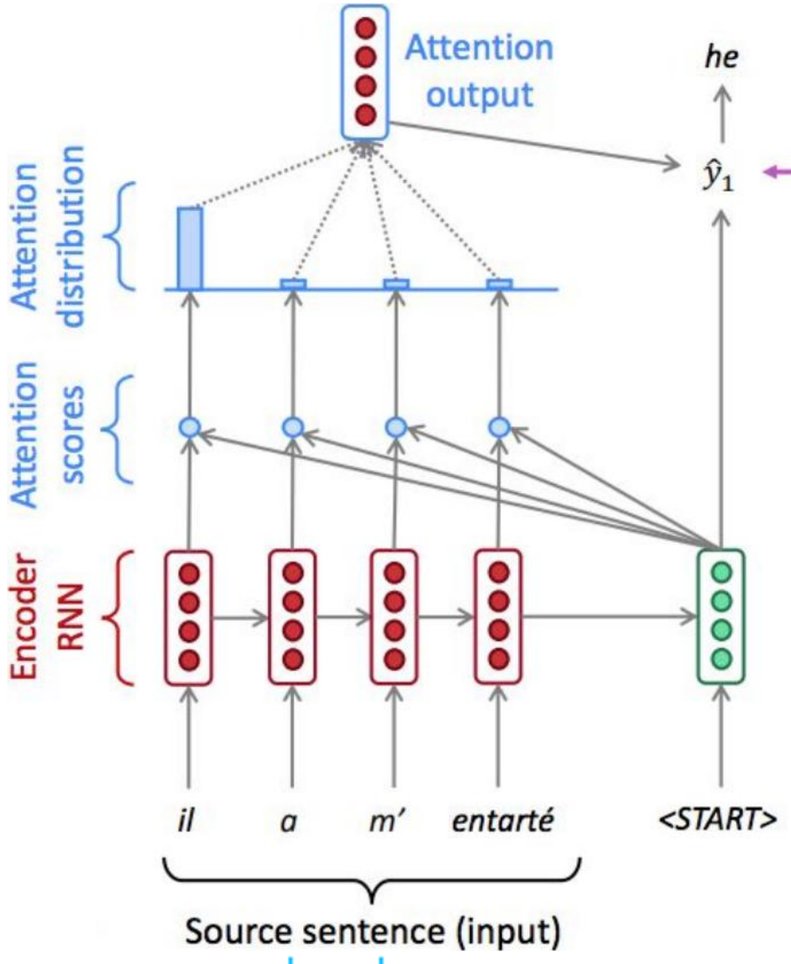
Attention *Mechanism*



Utilice la distribución de atención para obtener una **suma ponderada** de los estados ocultos del codificador.

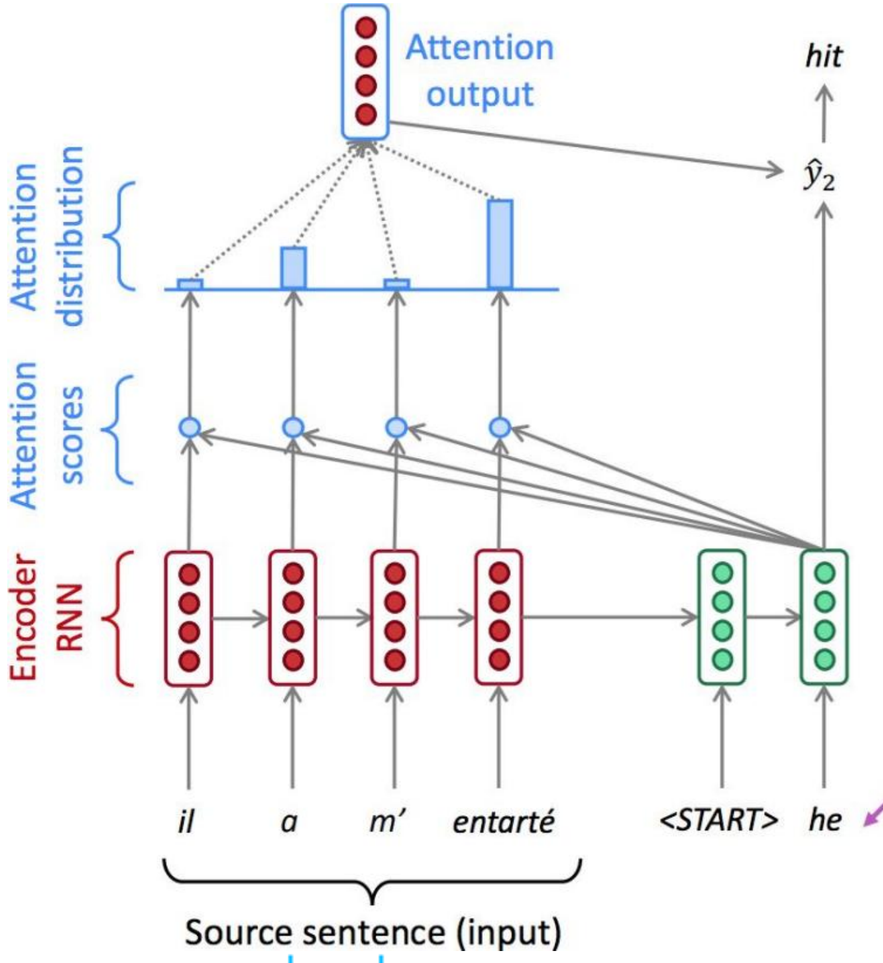
La salida de atención contiene principalmente información de los estados ocultos que recibieron mucha atención.

Attention *Mechanism*



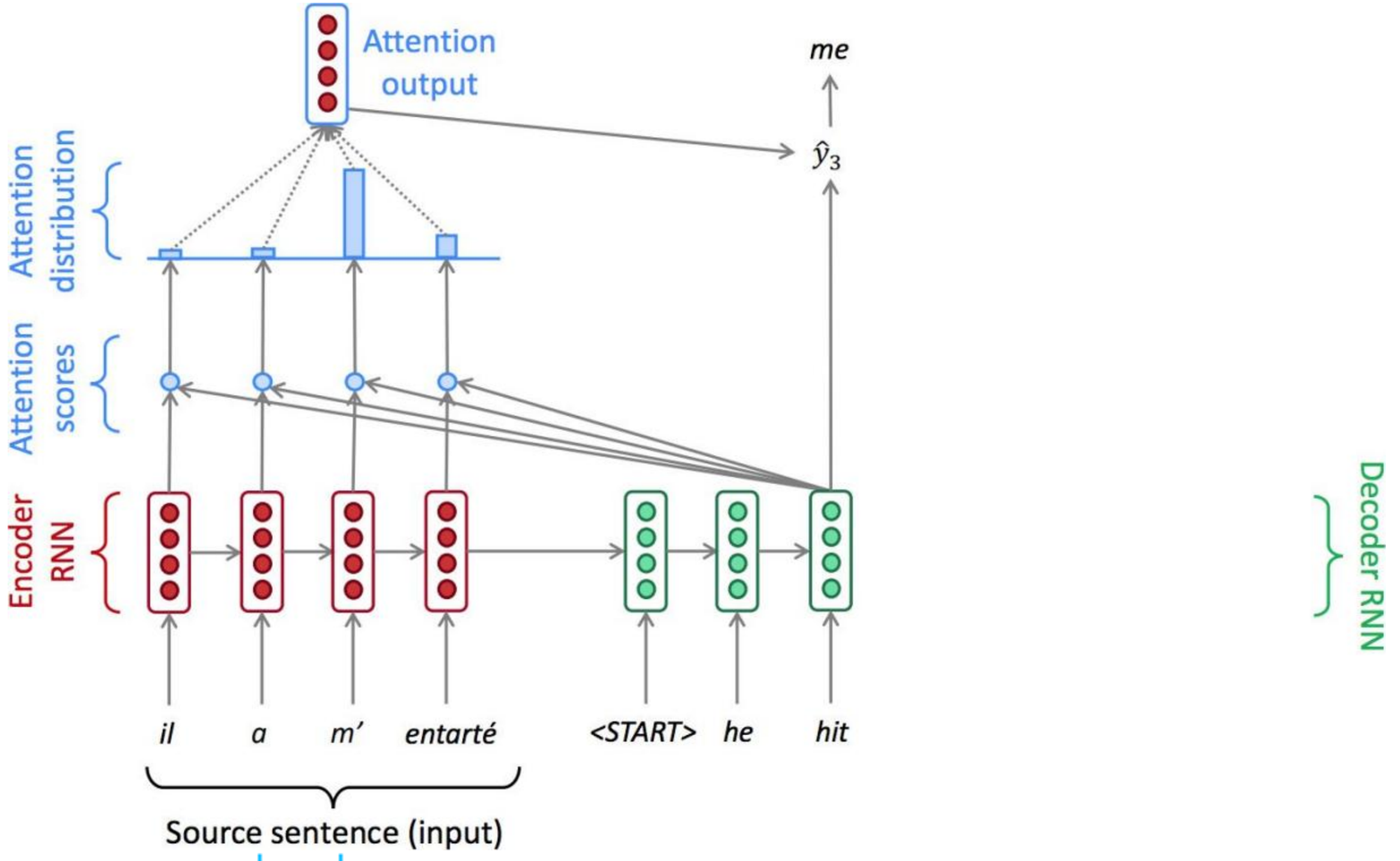
Concatenar la salida de atención con el estado oculto del decodificador, luego utilícelo para calcular $\hat{y}_1$ como antes.

Attention *Mechanism*

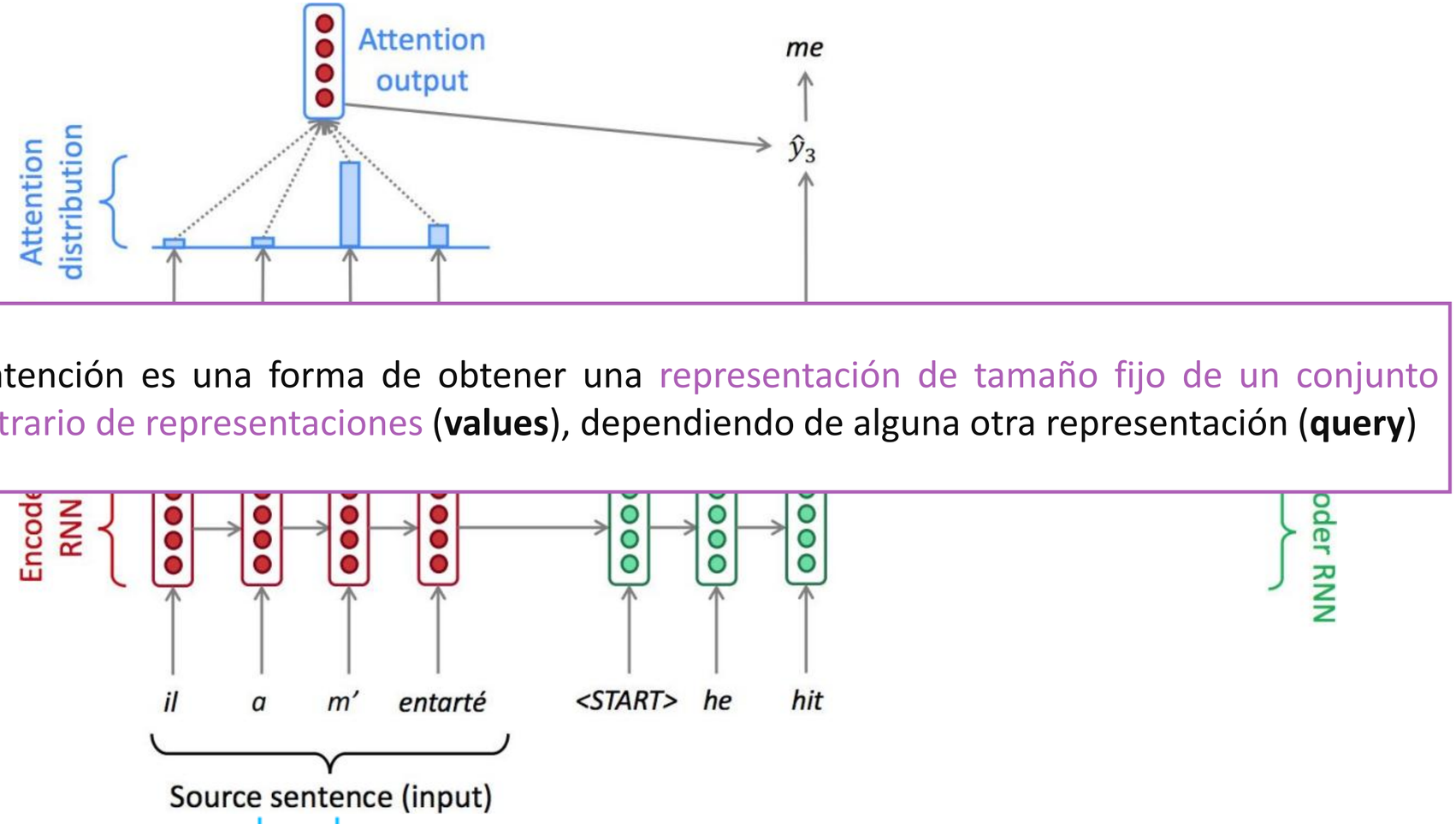


A veces tomamos la salida de atención del paso anterior y también la alimentamos al decodificador (junto con la entrada habitual del decodificador).

Attention *Mechanism*

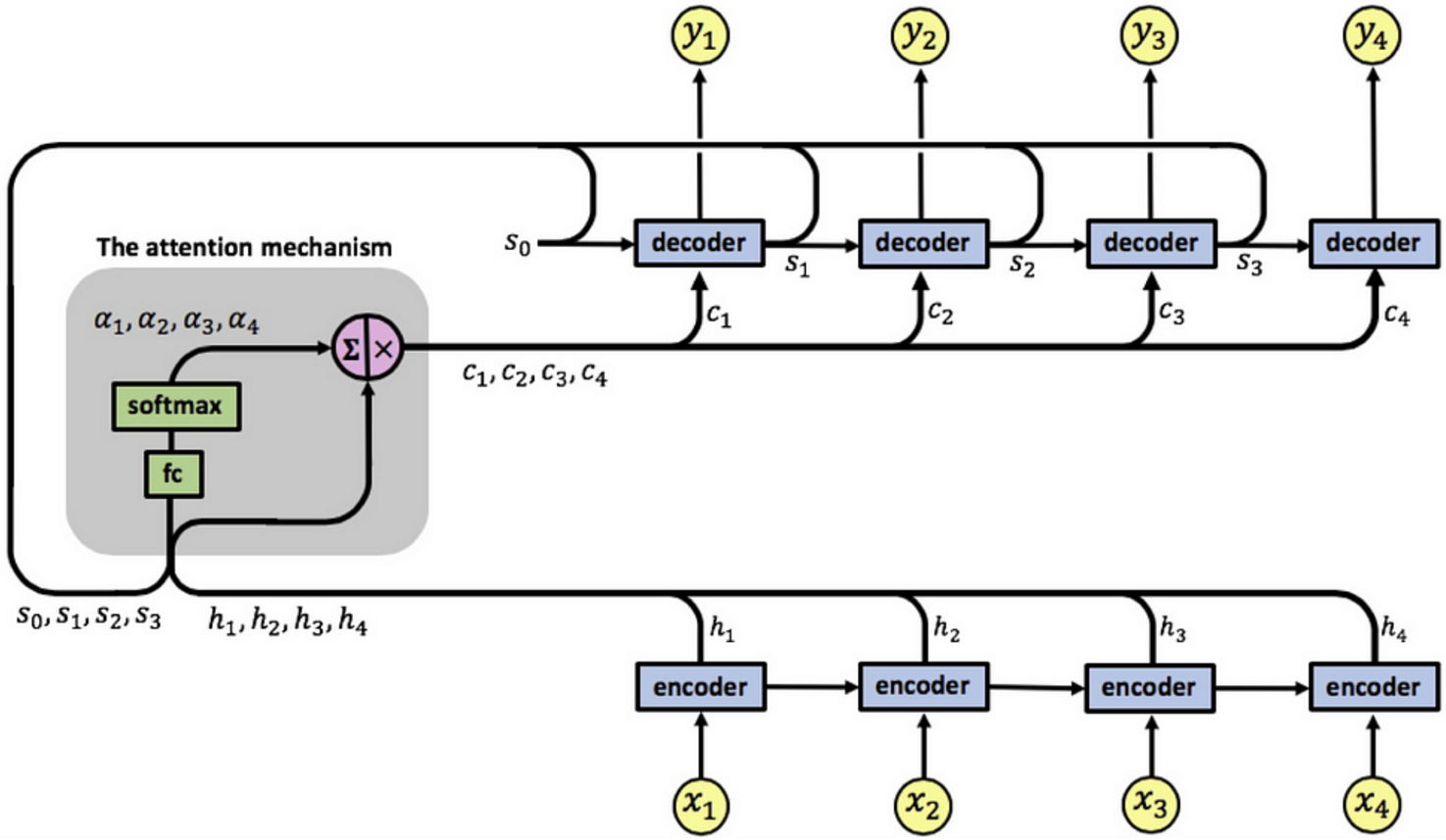


Attention *Mechanism*



La atención es una forma de obtener una representación de tamaño fijo de un conjunto arbitrario de representaciones (**values**), dependiendo de alguna otra representación (**query**)

Attention *Mechanism*



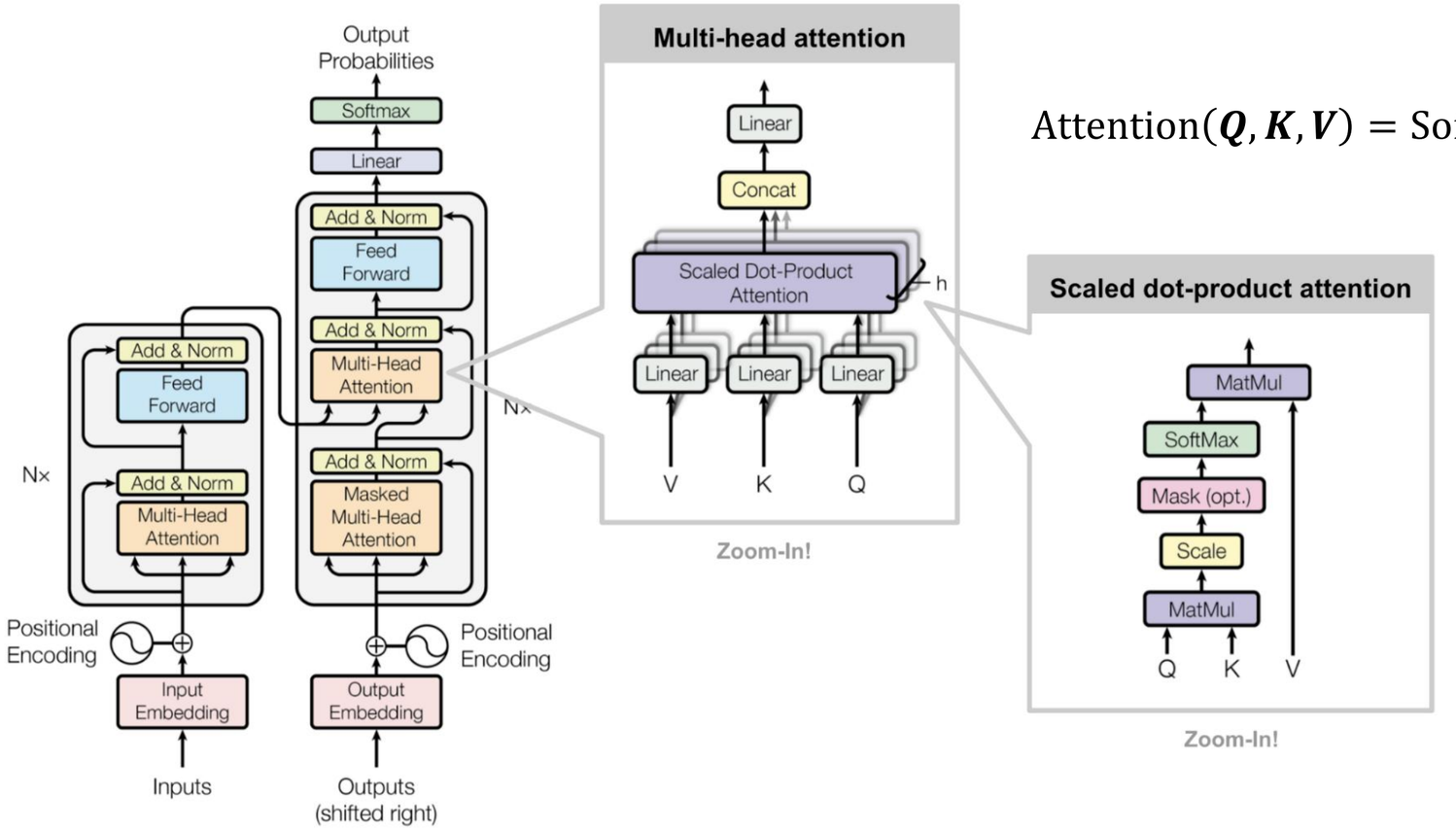
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

2.

Self-*Attention*

TRANSFORMATEC

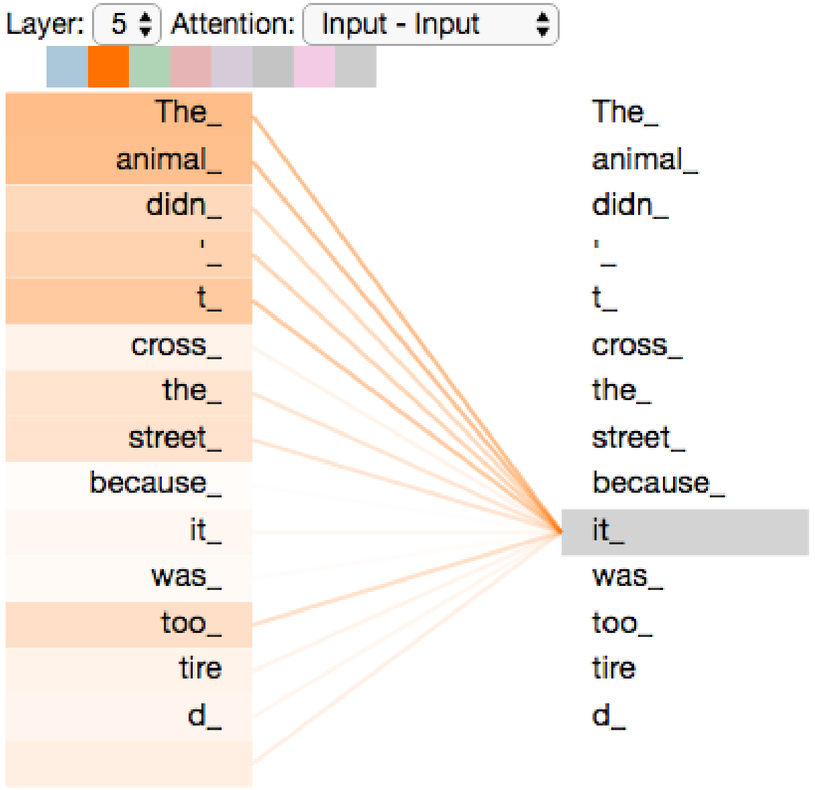
Attention is *all you need*



$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V}$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



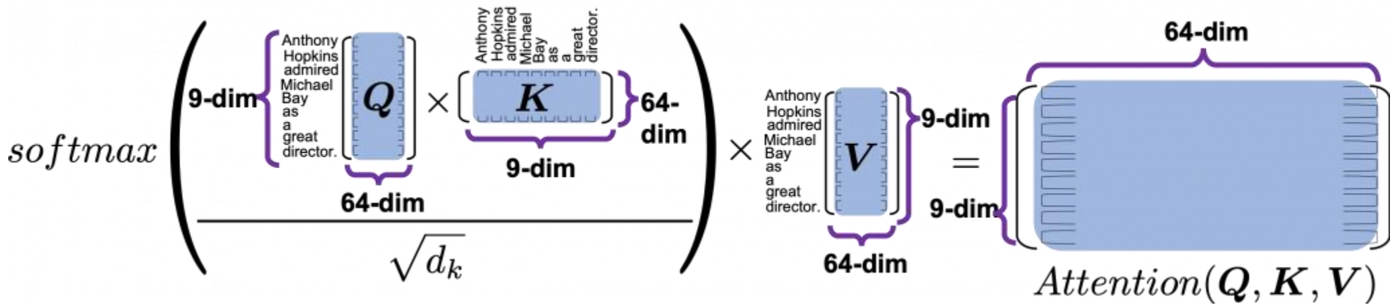
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Key (K) Actúa como un identificador de cada entrada. Es lo que el query usa para decidir qué tan relevante es una entrada en específico.

Query (Q) Representa lo que esta buscando. Se genera para cada entrada y se contra los keys de los otros elementos de la secuencia.

Value (V) Contiene la información que será usada en la salida. Una vez que se decida a quién prestar atención (a partir de queries y keys), ponderamos los values para generar la representación final.



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Key (K) Actúa como un identificador de cada entrada. Es lo que el query usa para decidir qué tan relevante es una entrada en específico.

Query (Q) Representa lo que esta buscando. Se genera para cada entrada y se contra los keys de los otros elementos de la secuencia.

Value (V) Contiene la información que será usada en la salida. Una vez que se decida a quién prestar atención (a partir de los queries y keys), ponderamos los values para generar la representación final.

Query (Q):
¿Qué acción está asociada con 'gato'?

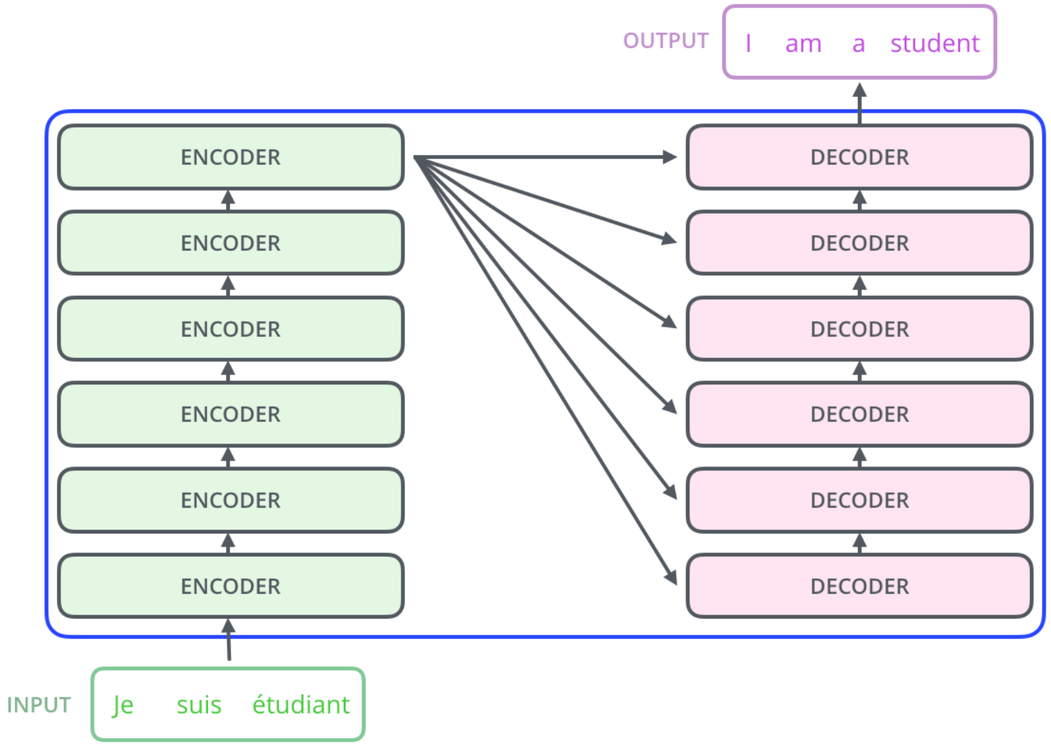
El **gato** *persigue* al ratón

Key (K):
Es un verbo de acción

Value (V):
Información semántica de la palabra

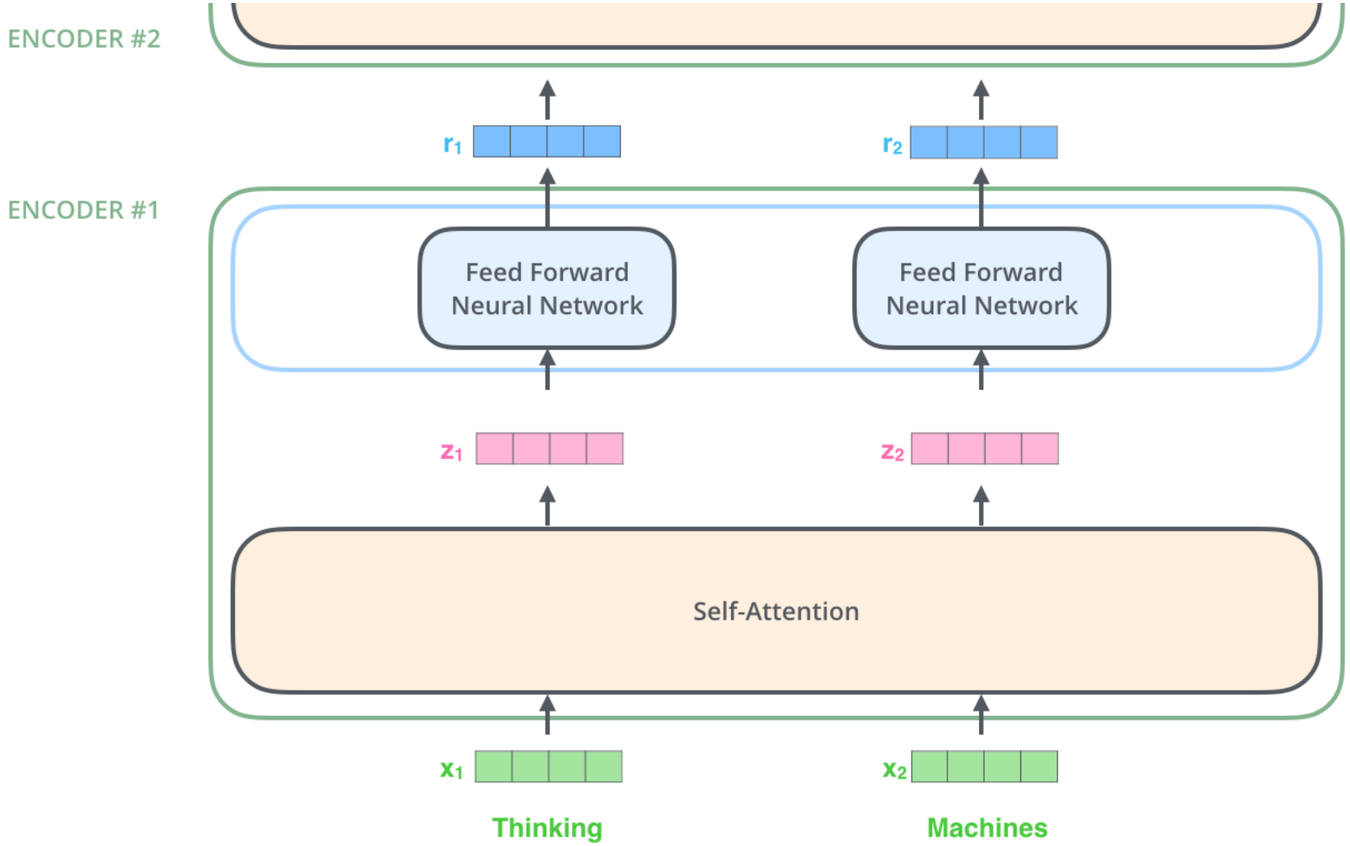
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



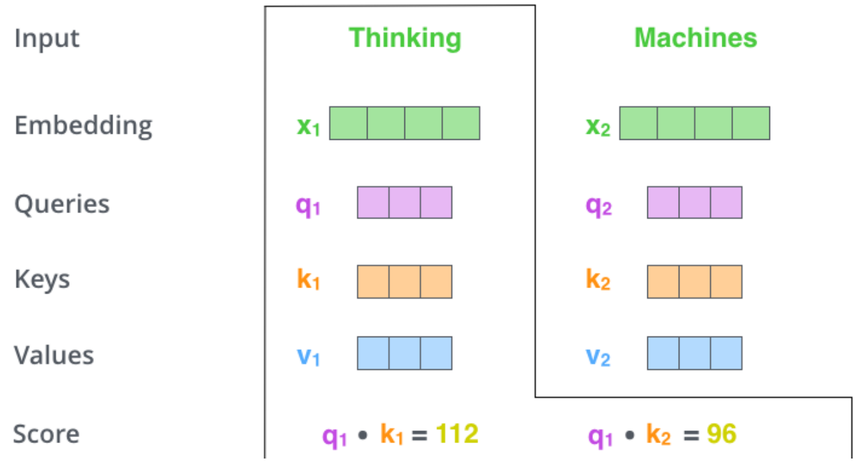
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



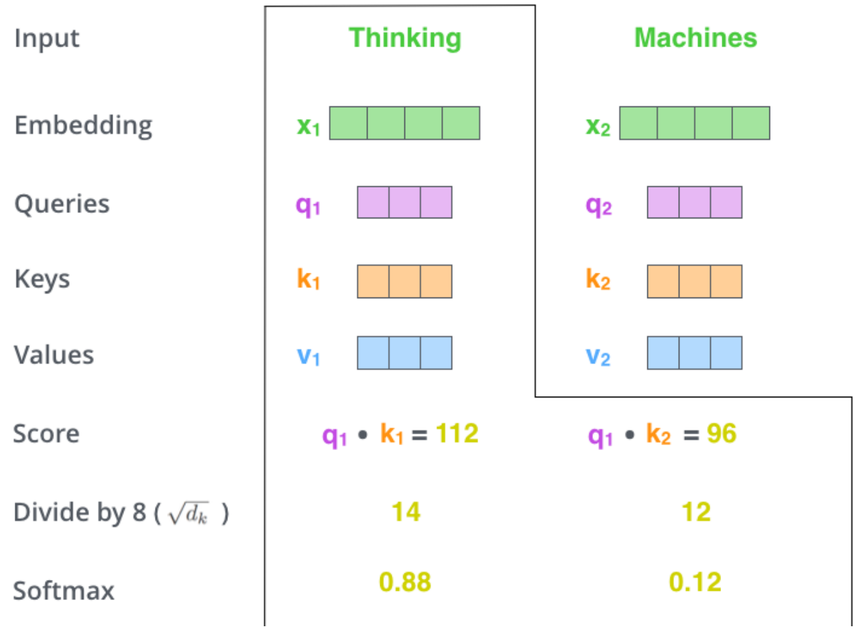
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



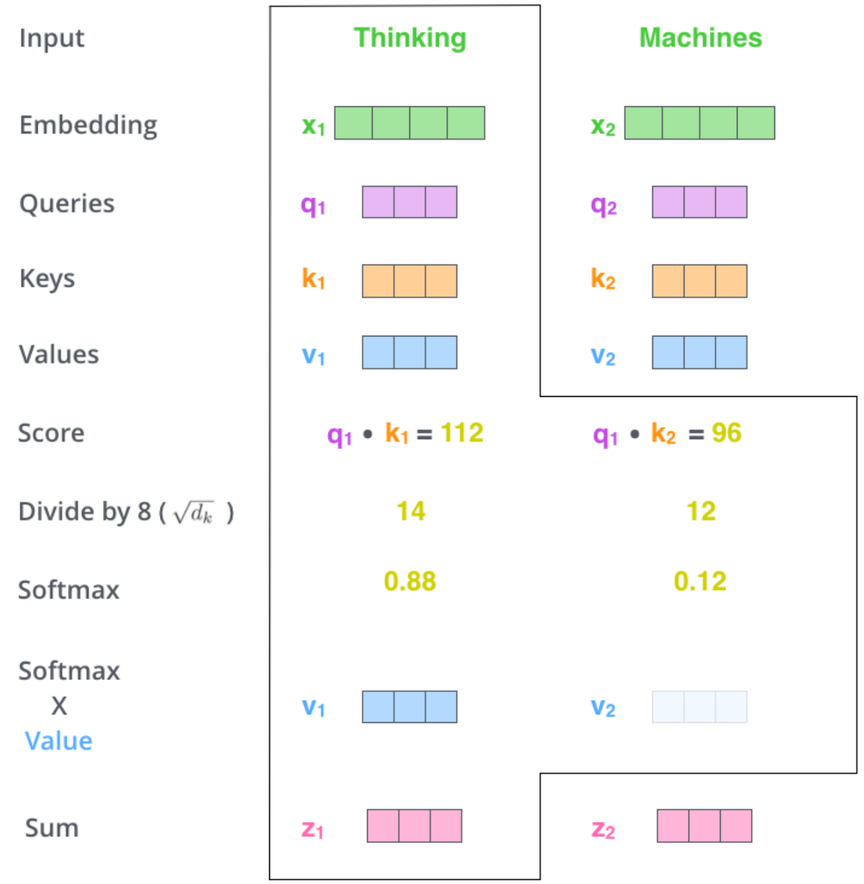
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



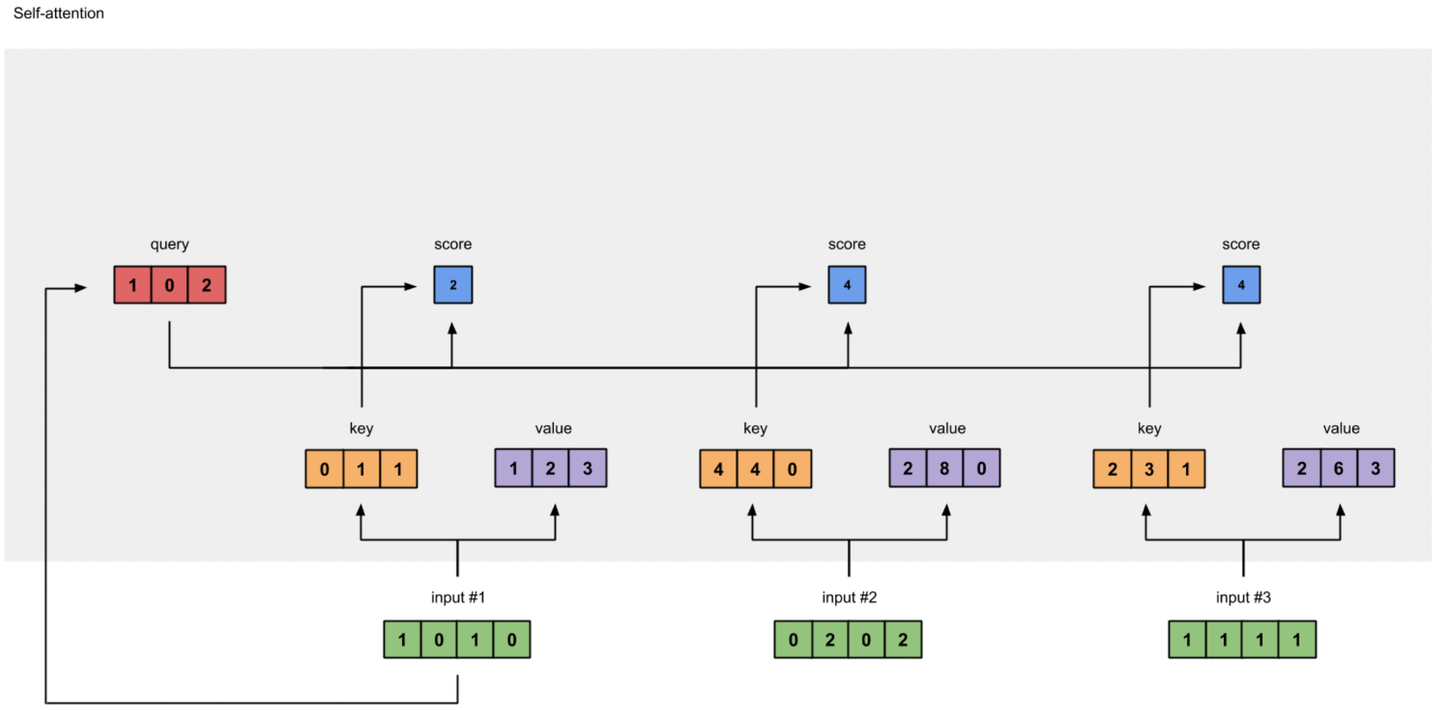
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



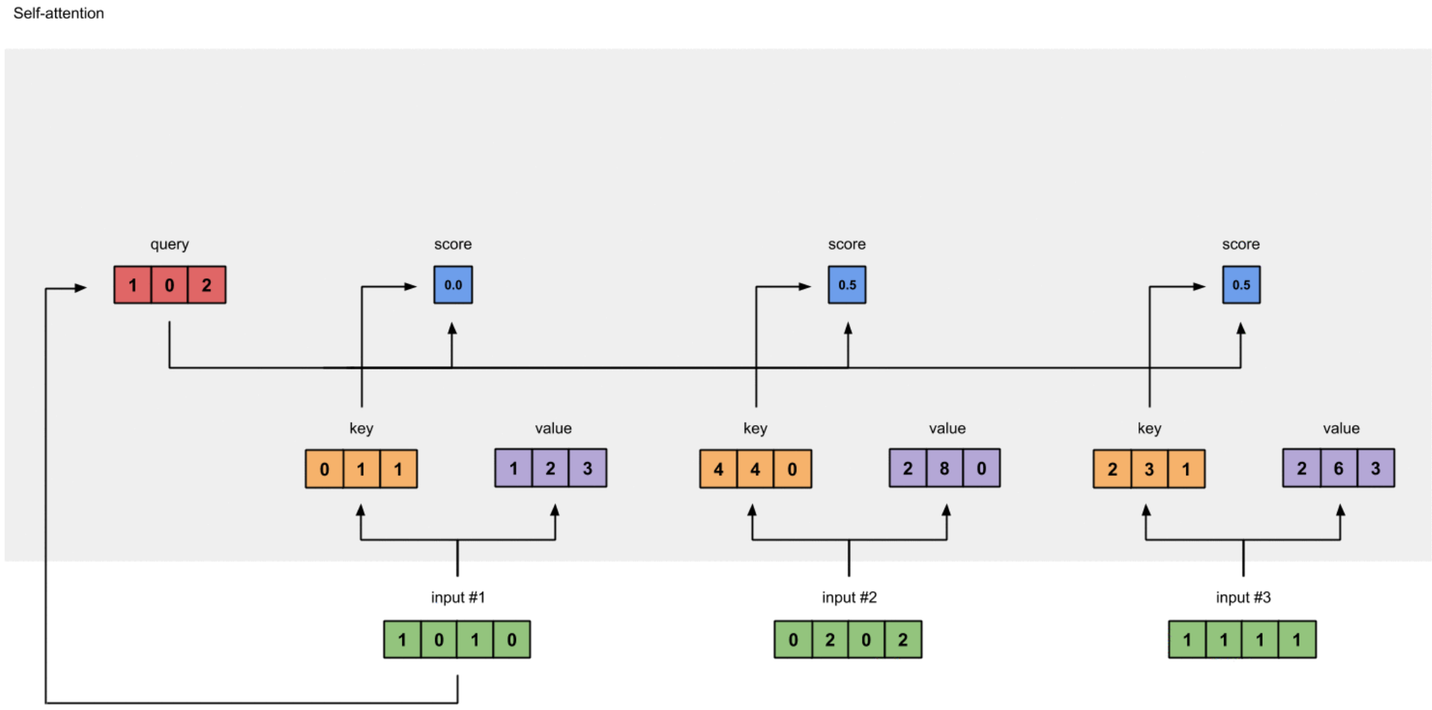
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



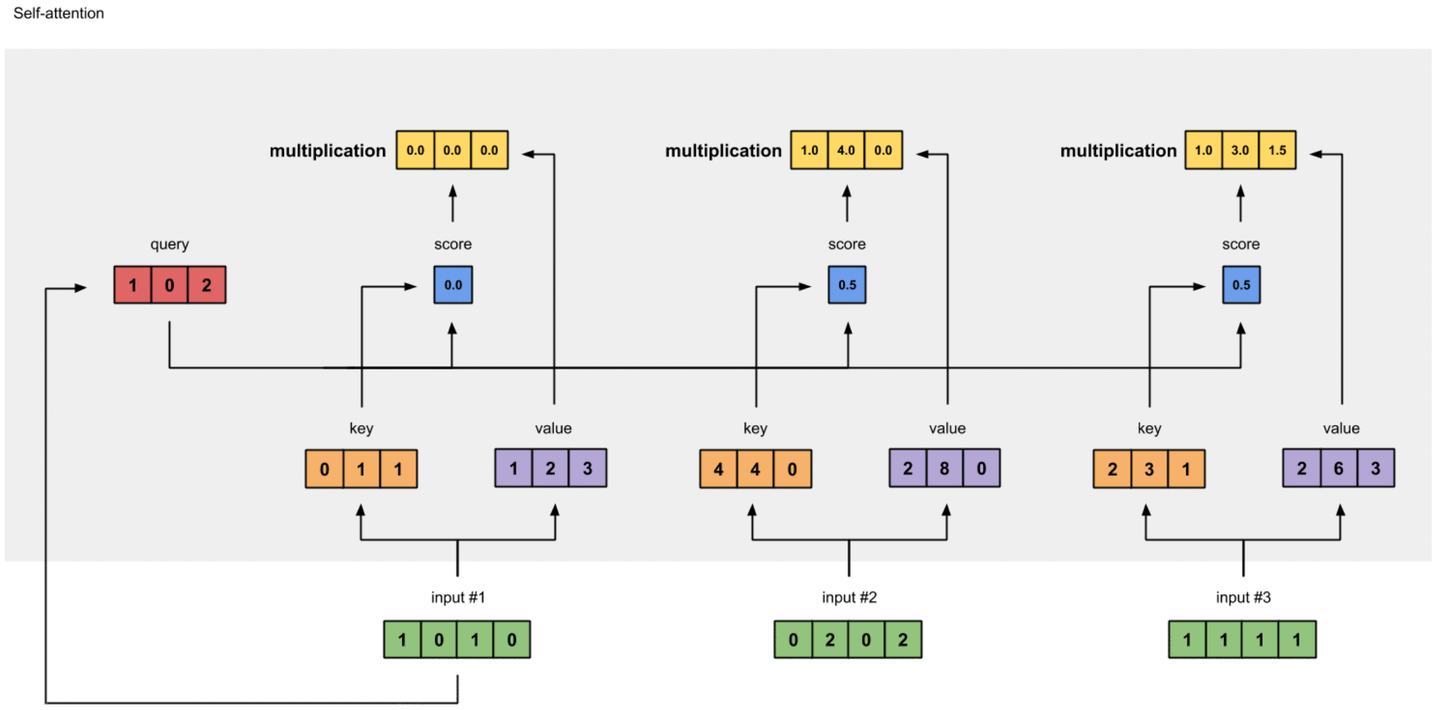
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



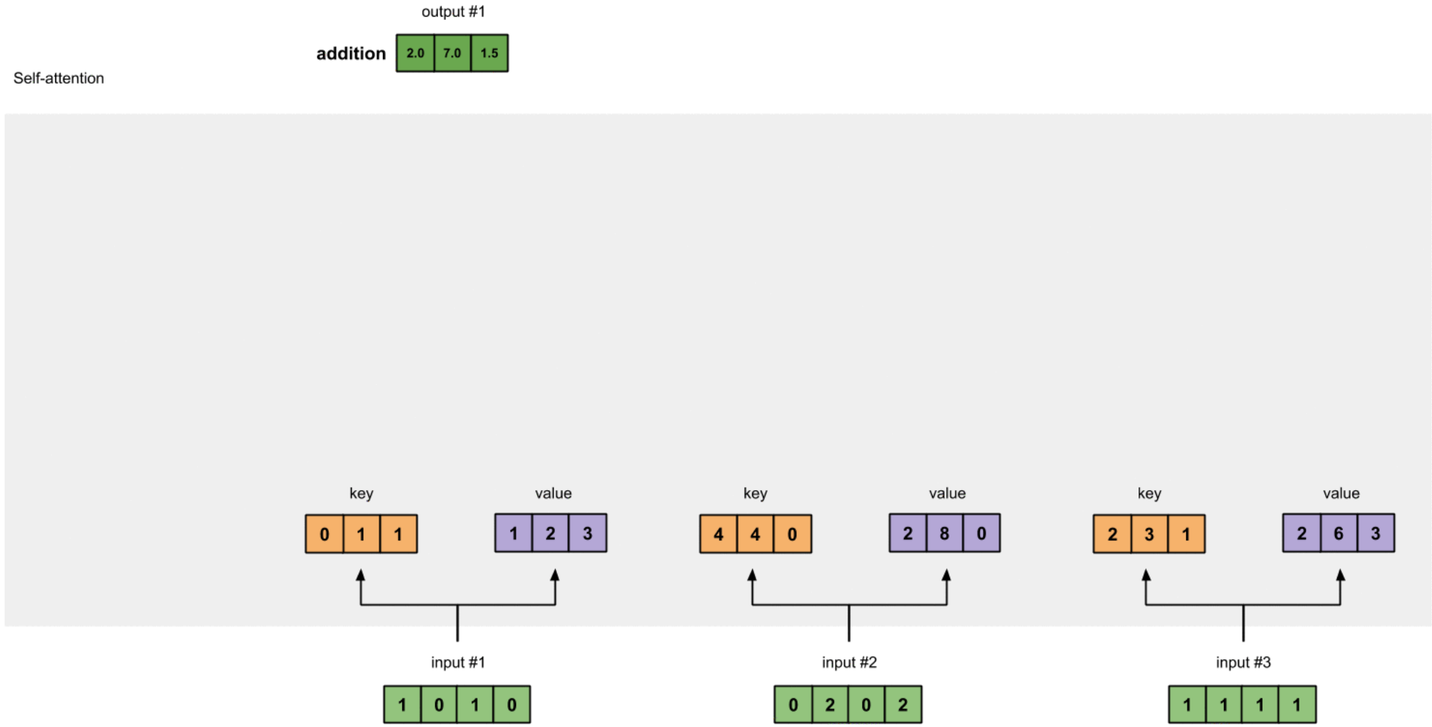
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



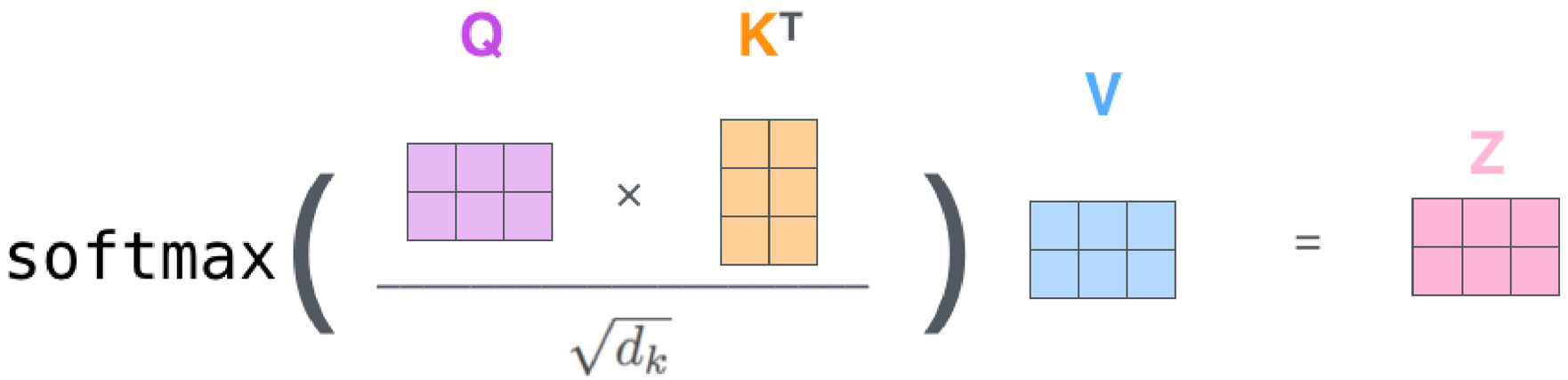
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}$$

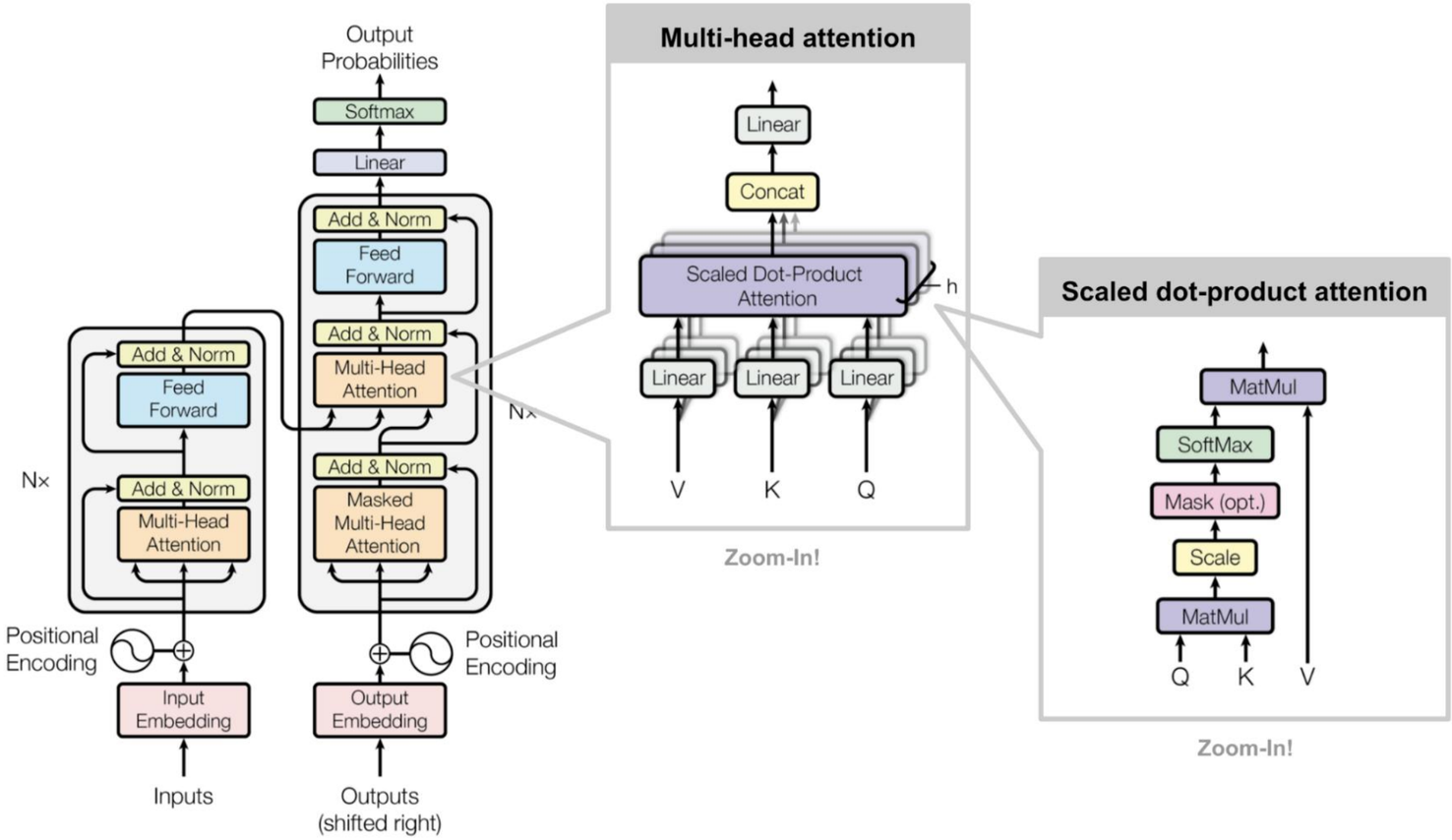
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



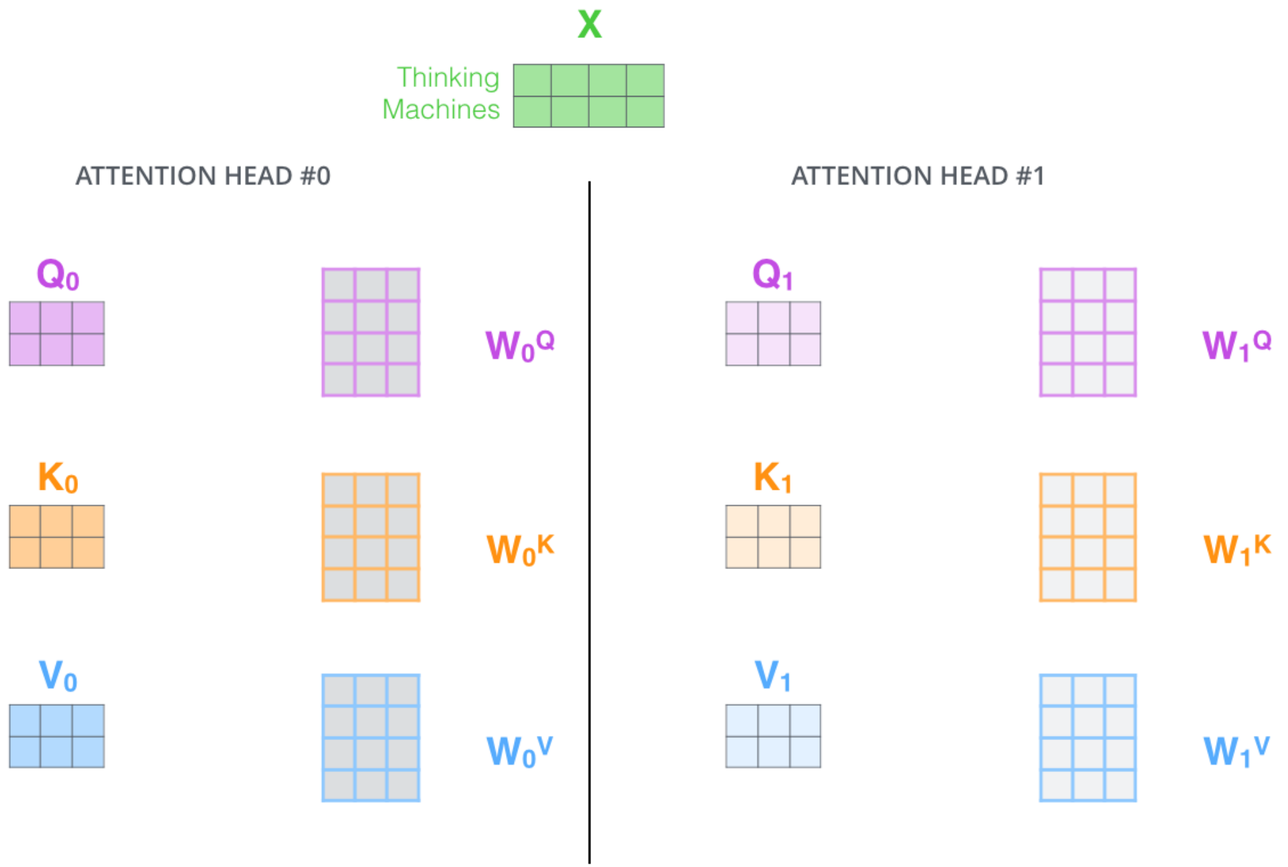
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



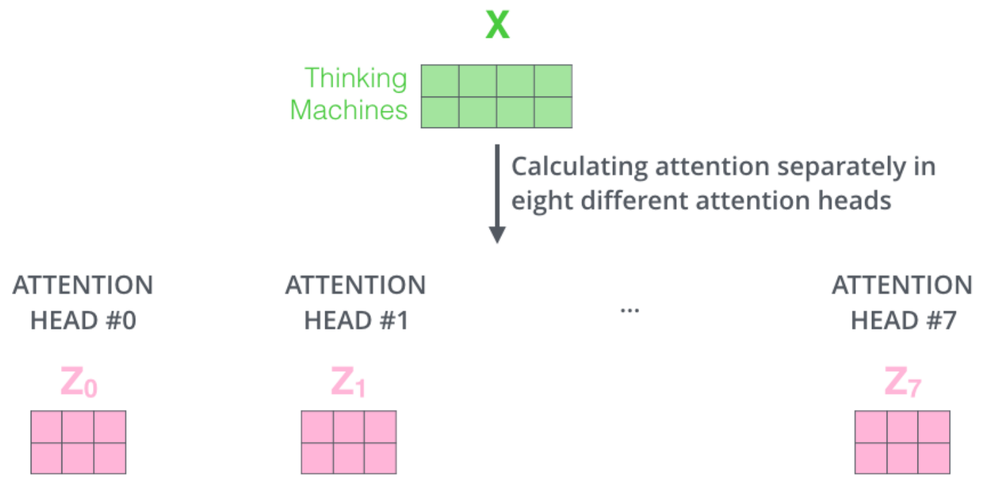
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

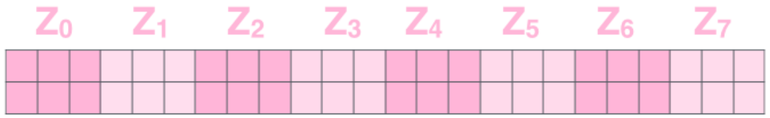
Attention is *all you need*



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

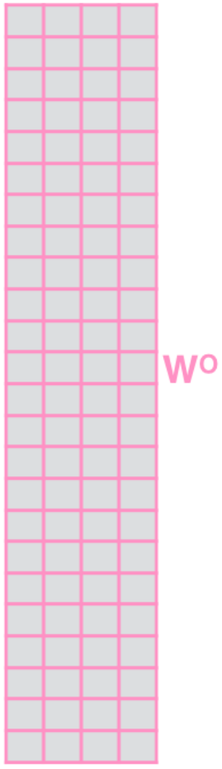
Attention is *all you need*

1) Concatenate all the attention heads



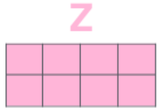
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

1) This is our input sentence*

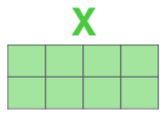
2) We embed each word*

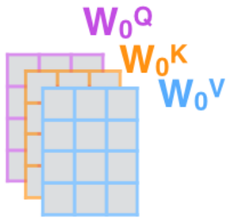
3) Split into 8 heads. We multiply X or R with weight matrices

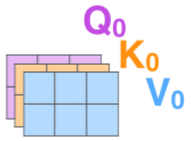
4) Calculate attention using the resulting Q/K/V matrices

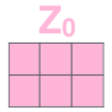
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

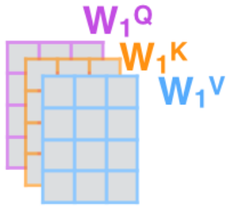
Thinking Machines

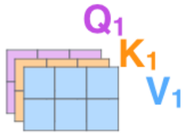


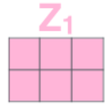






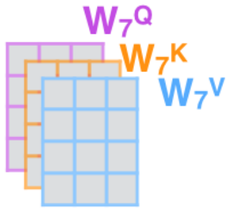




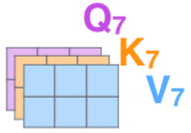


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

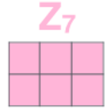
...

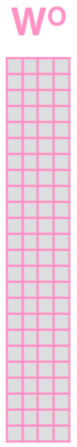


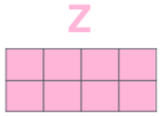
...



...



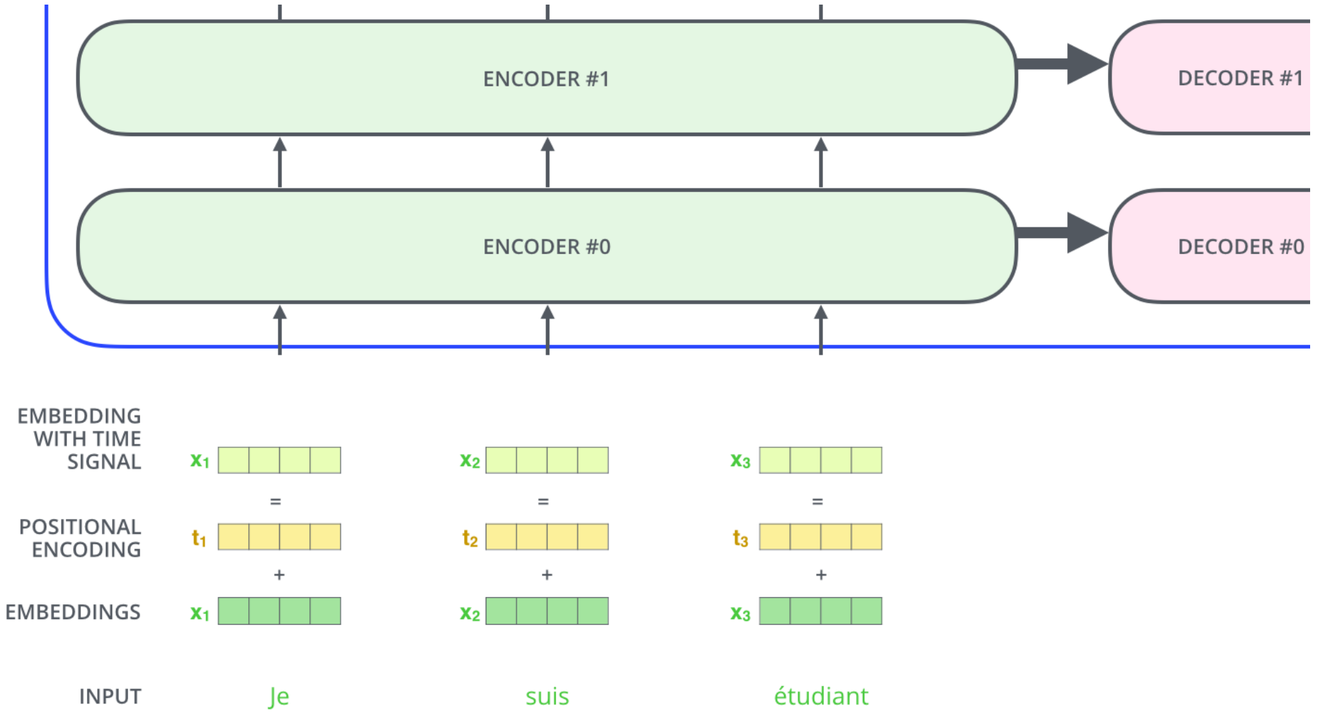




Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

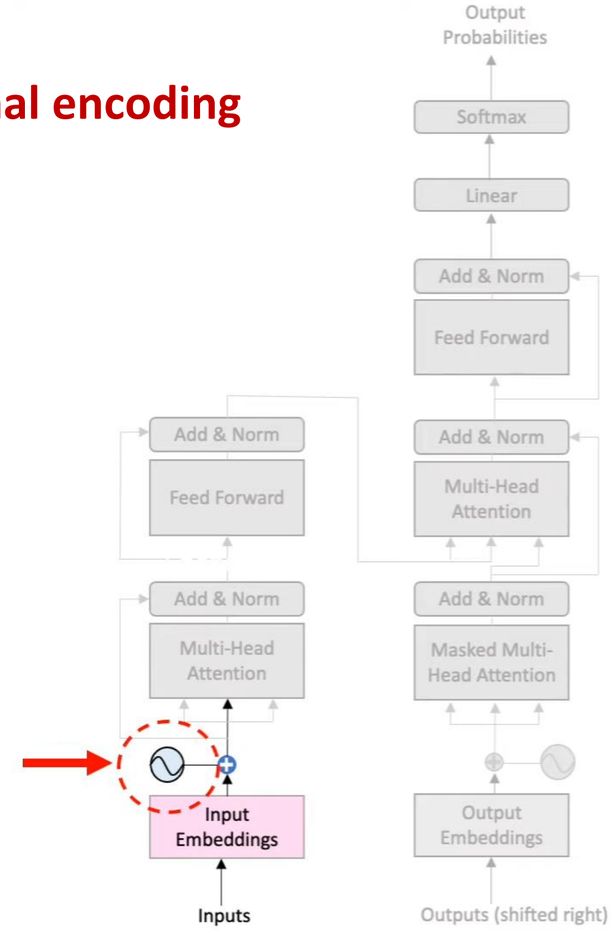
Positional encoding



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding



e_0	p_0	e_1	p_2	e_2	p_1	e_3	p_3
0.42	?	0.87	?	0.02	?	0.02	?
0.31	?	-0.64	?	0.01	?	0.01	?
0.73	?	0.81	?	-0.24	?	-0.24	?
0.36	?	0.26	?	-0.07	?	-0.07	?
0.99	?	-0.35	?	0.00	?	0.00	?

Vaswani et al 2017

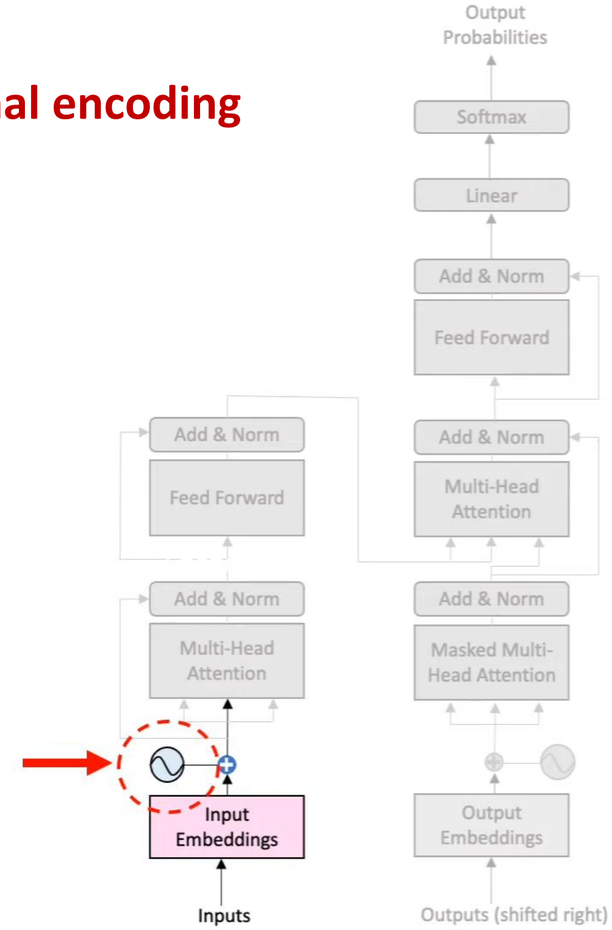
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

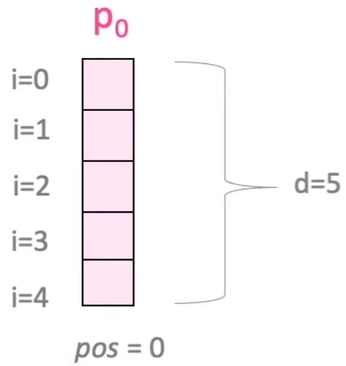
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding





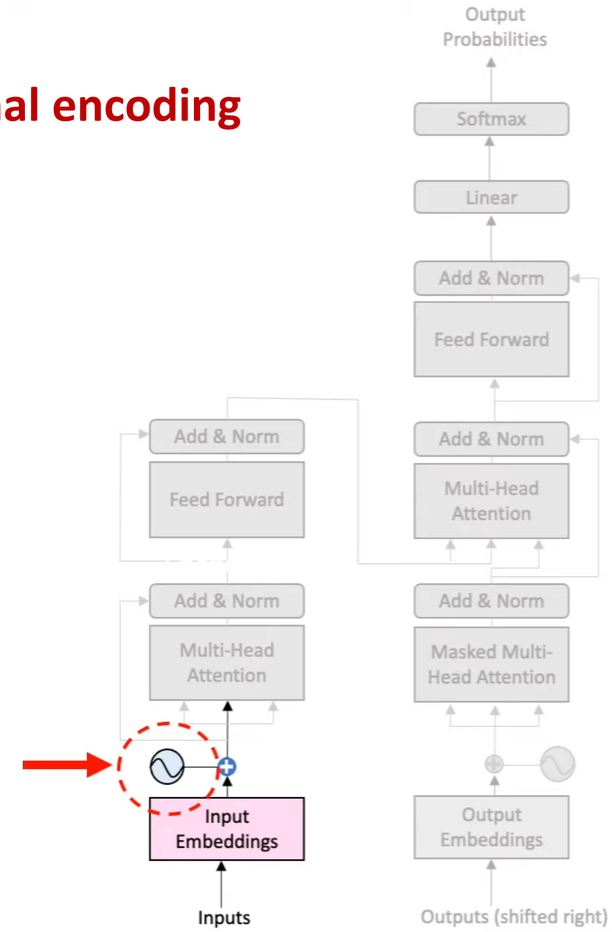
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

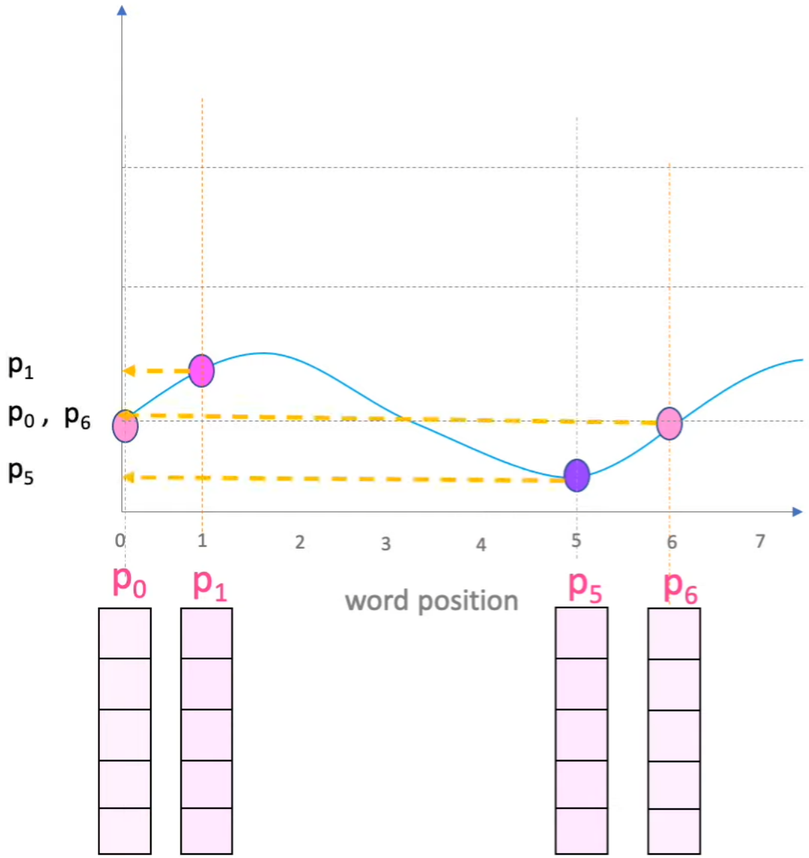
TRANSFORMATEC

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding

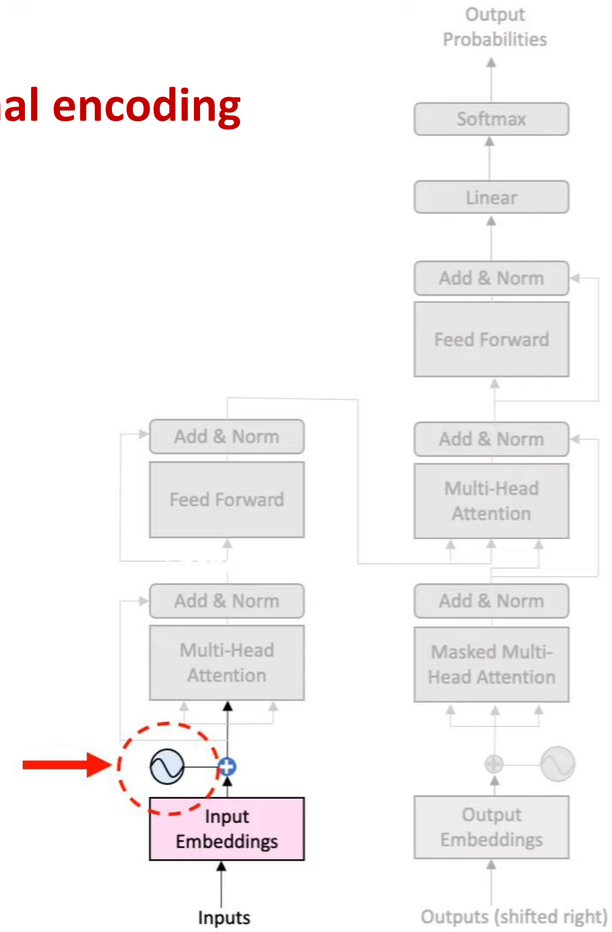


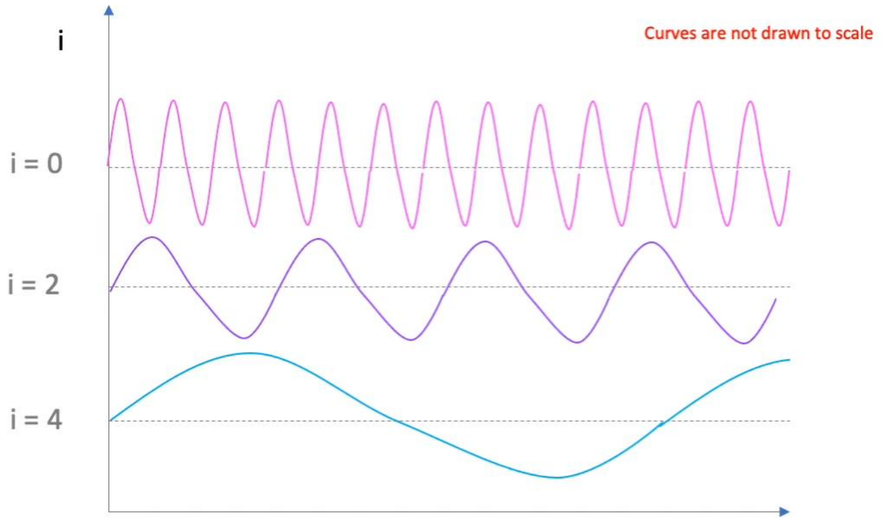


Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding



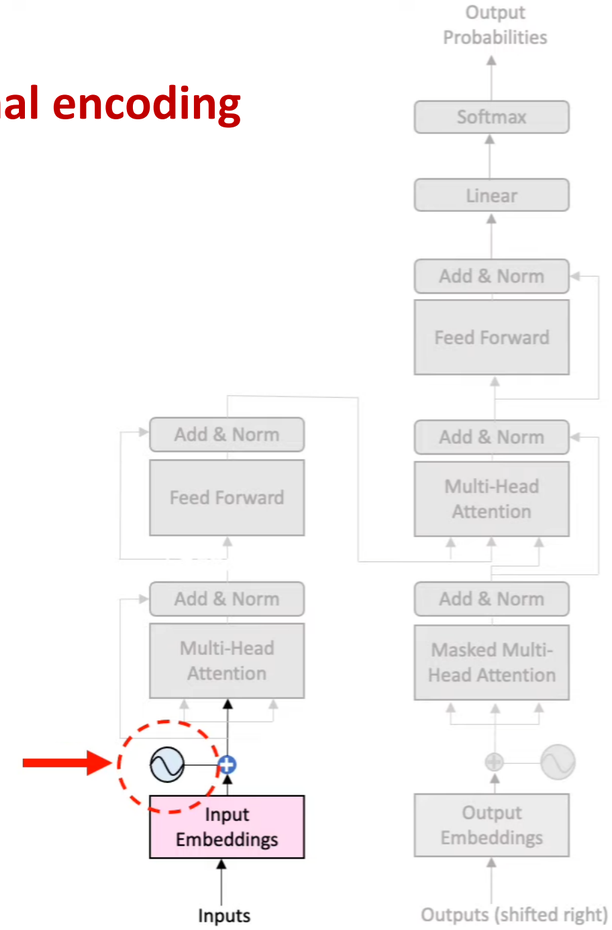


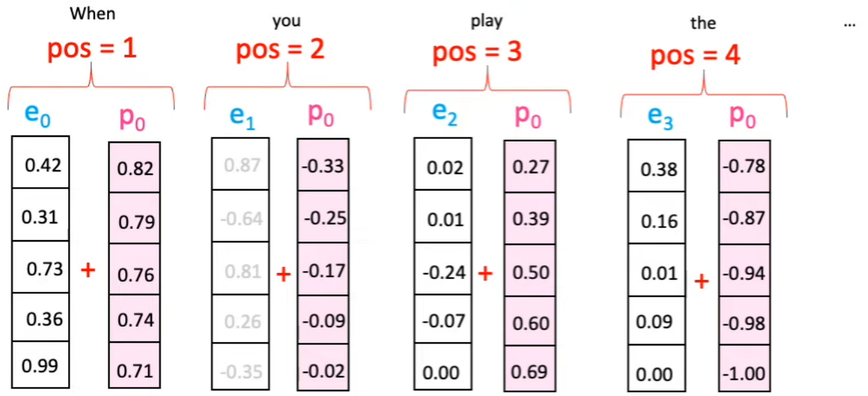
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding





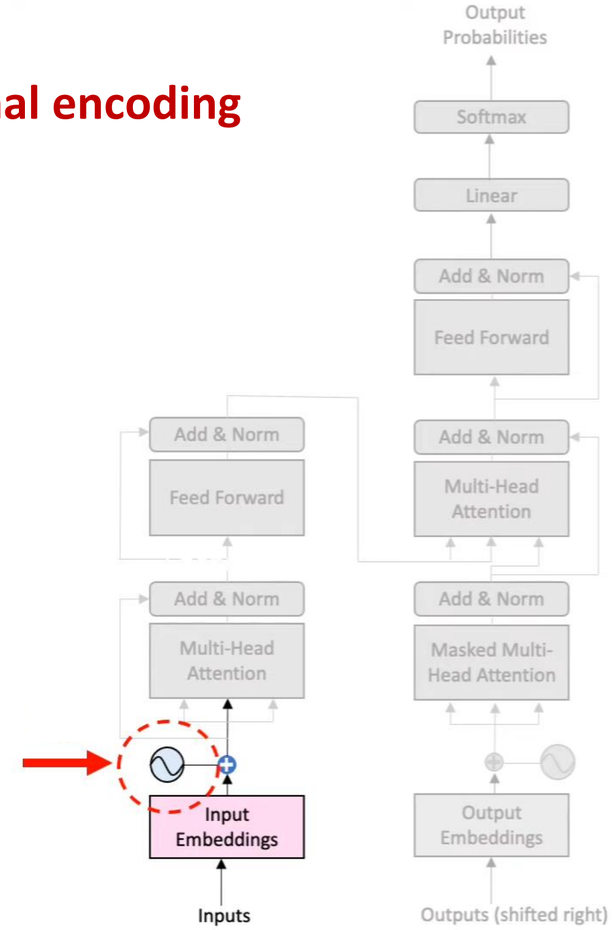
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

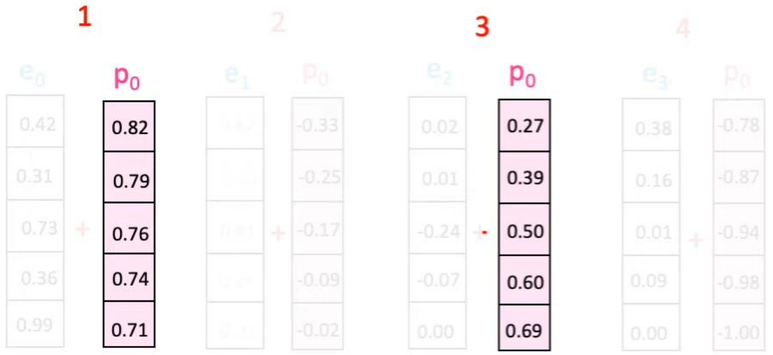
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding





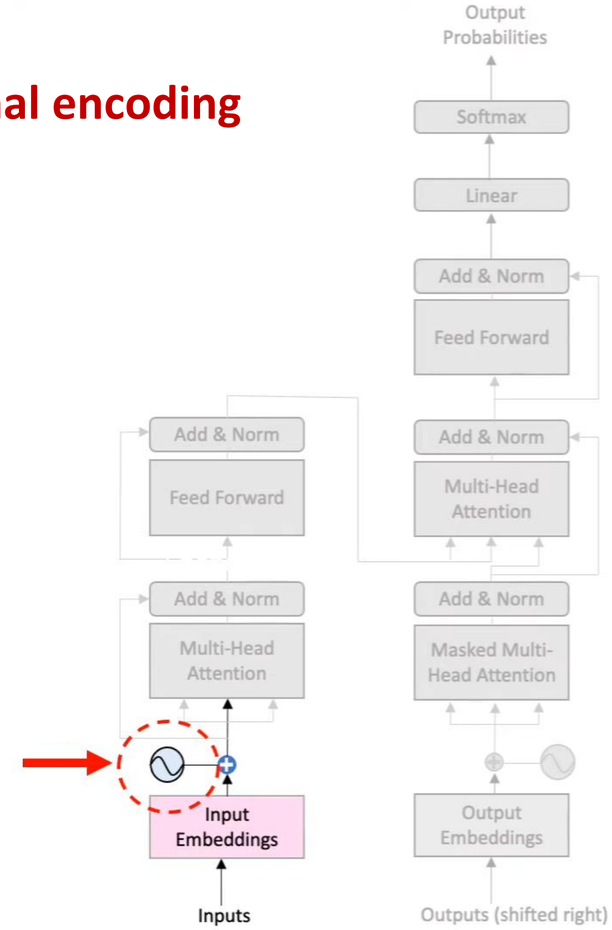
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding



$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

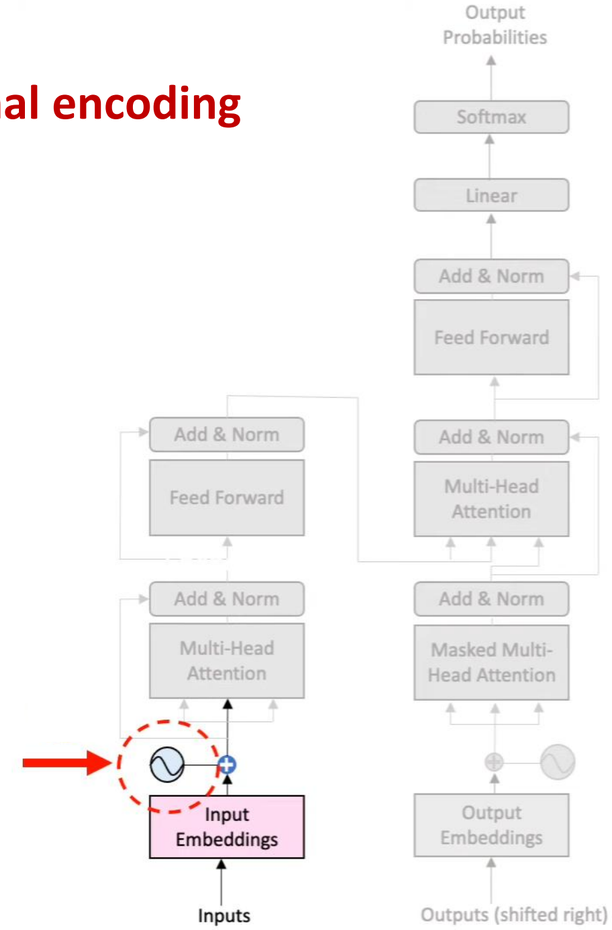
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

TRANSFORMATEC

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*

Positional encoding



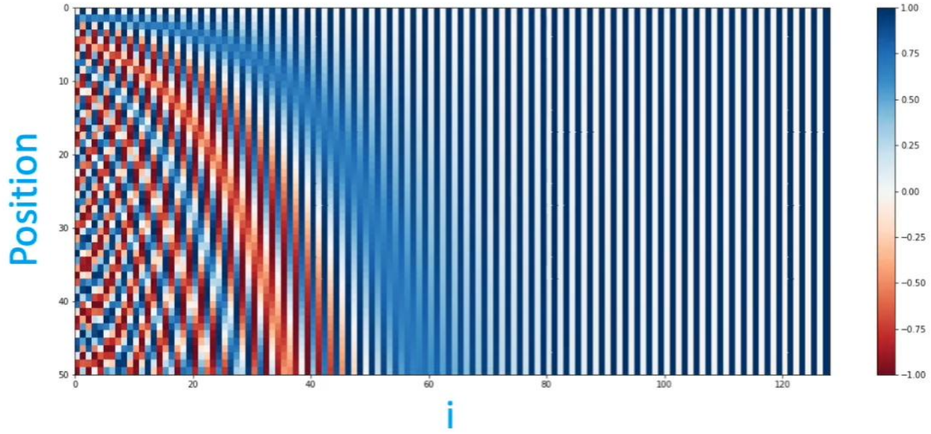


Image Source: Amirhoussein Kazemnejad blog-post *Transformer Architecture: The Positional Encoding*

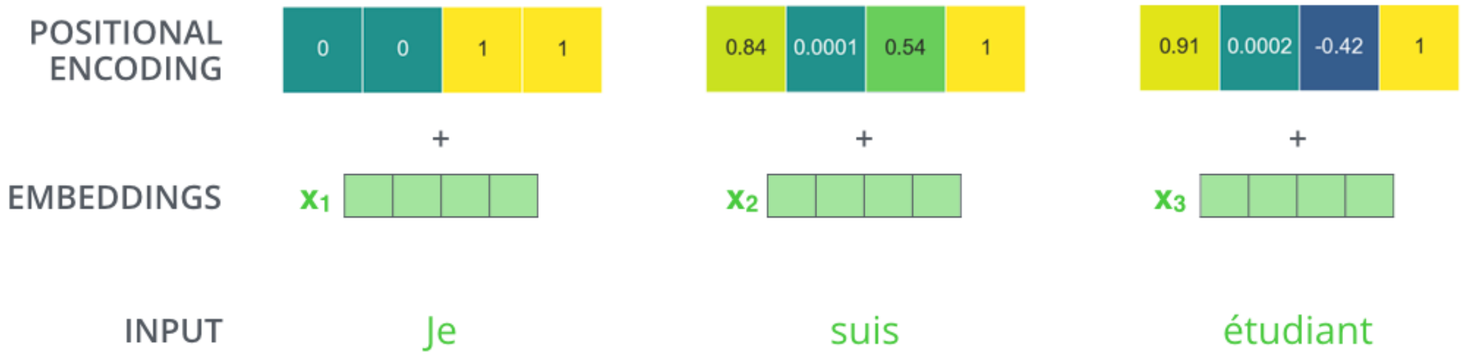
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

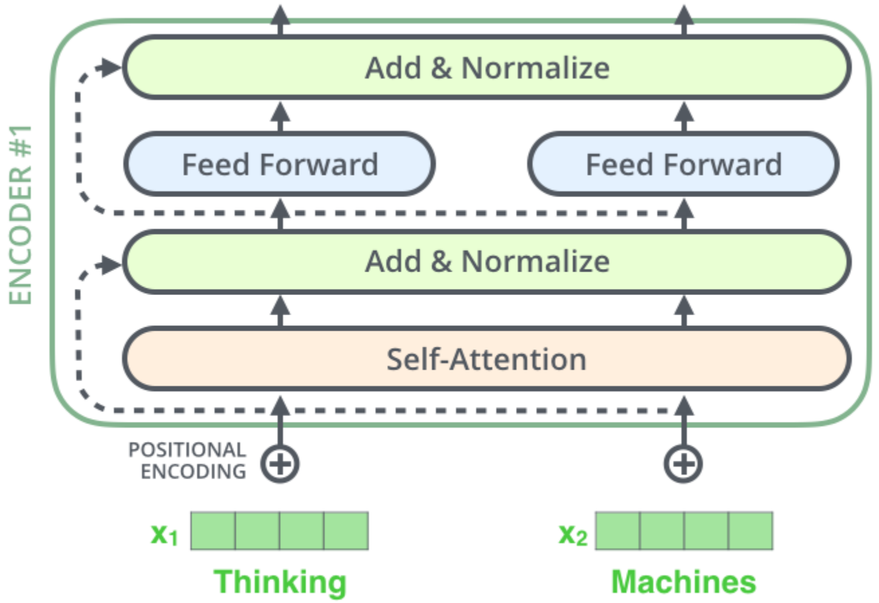
Attention is *all you need*

Positional encoding



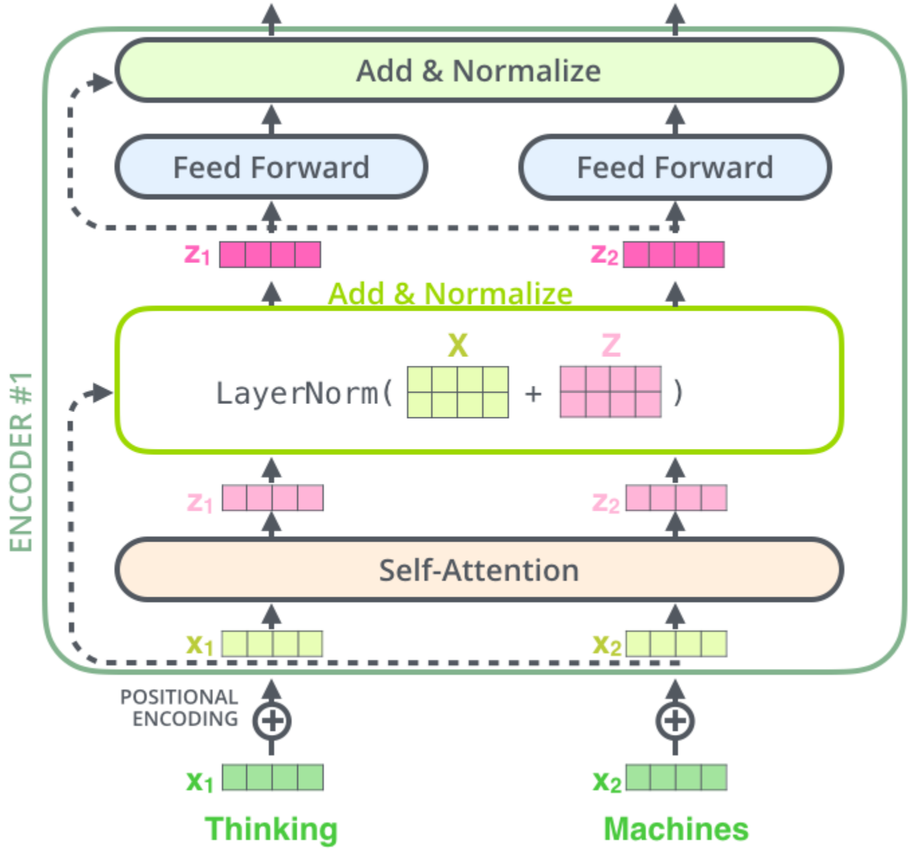
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



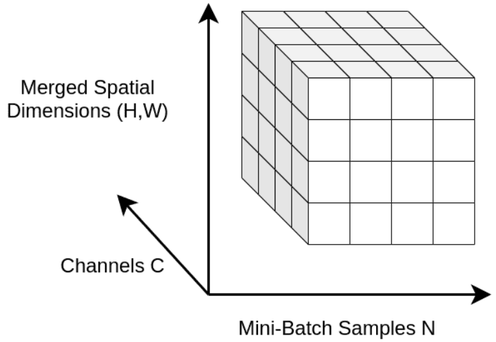
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

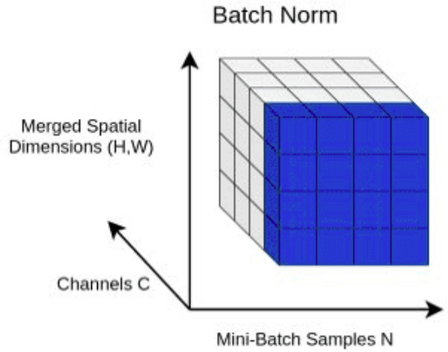
Attention is *all you need*



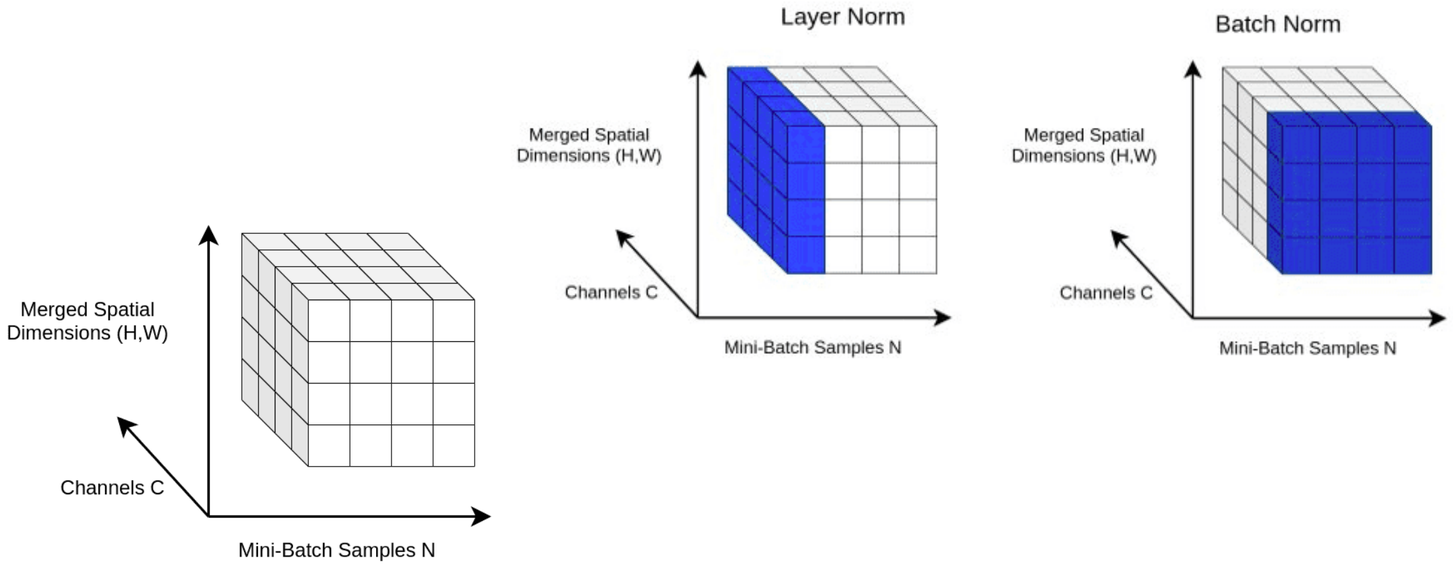
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

In-layer normalization techniques

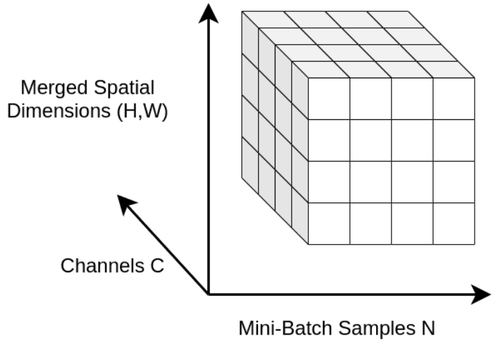


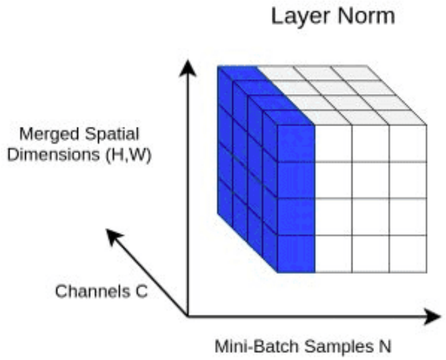


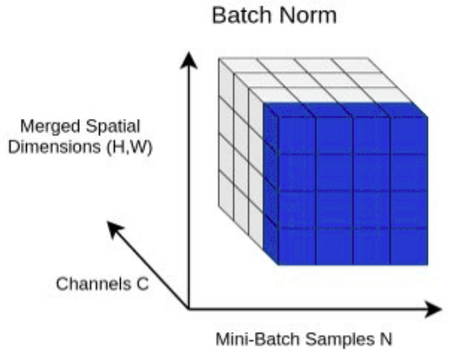
In-layer normalization techniques



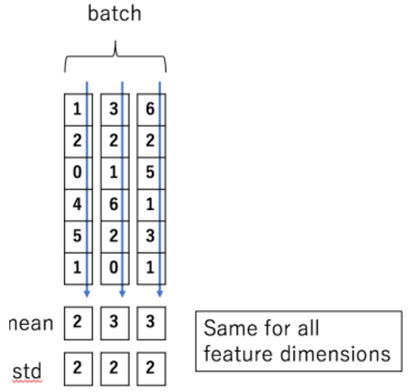
In-layer normalization techniques



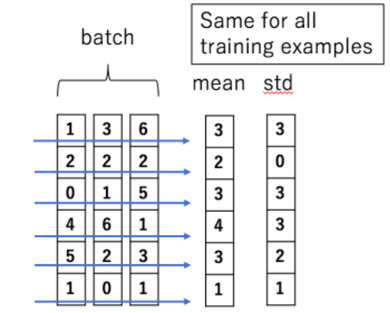




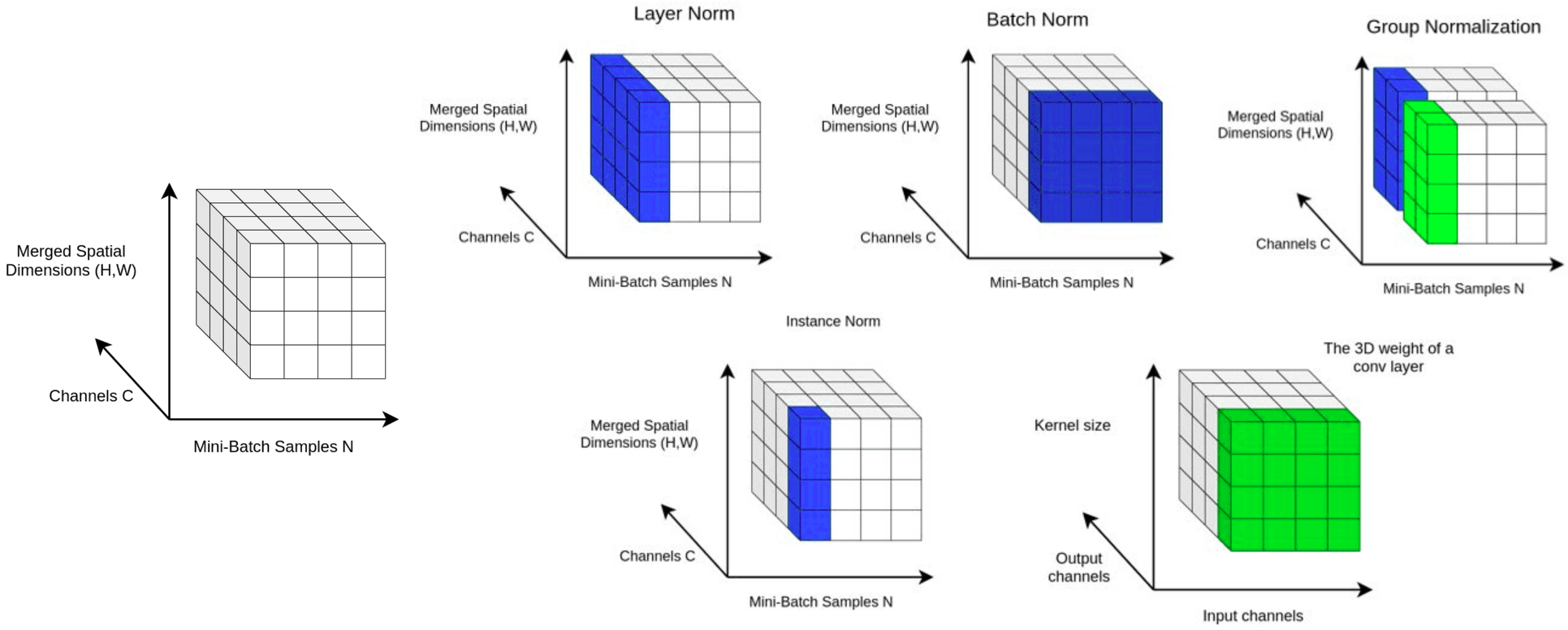
Layer Normalization



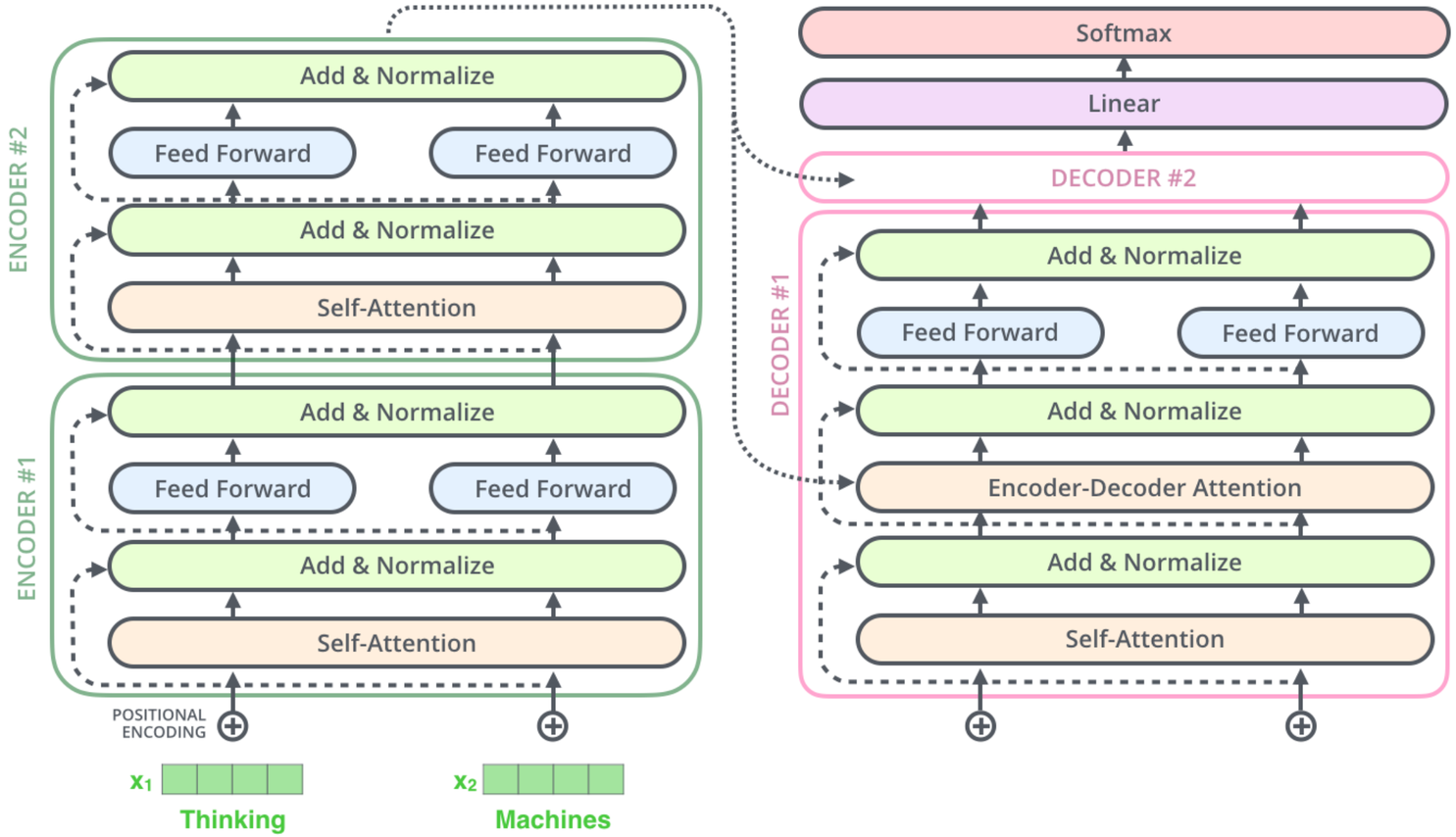
Batch Normalization



In-layer normalization techniques

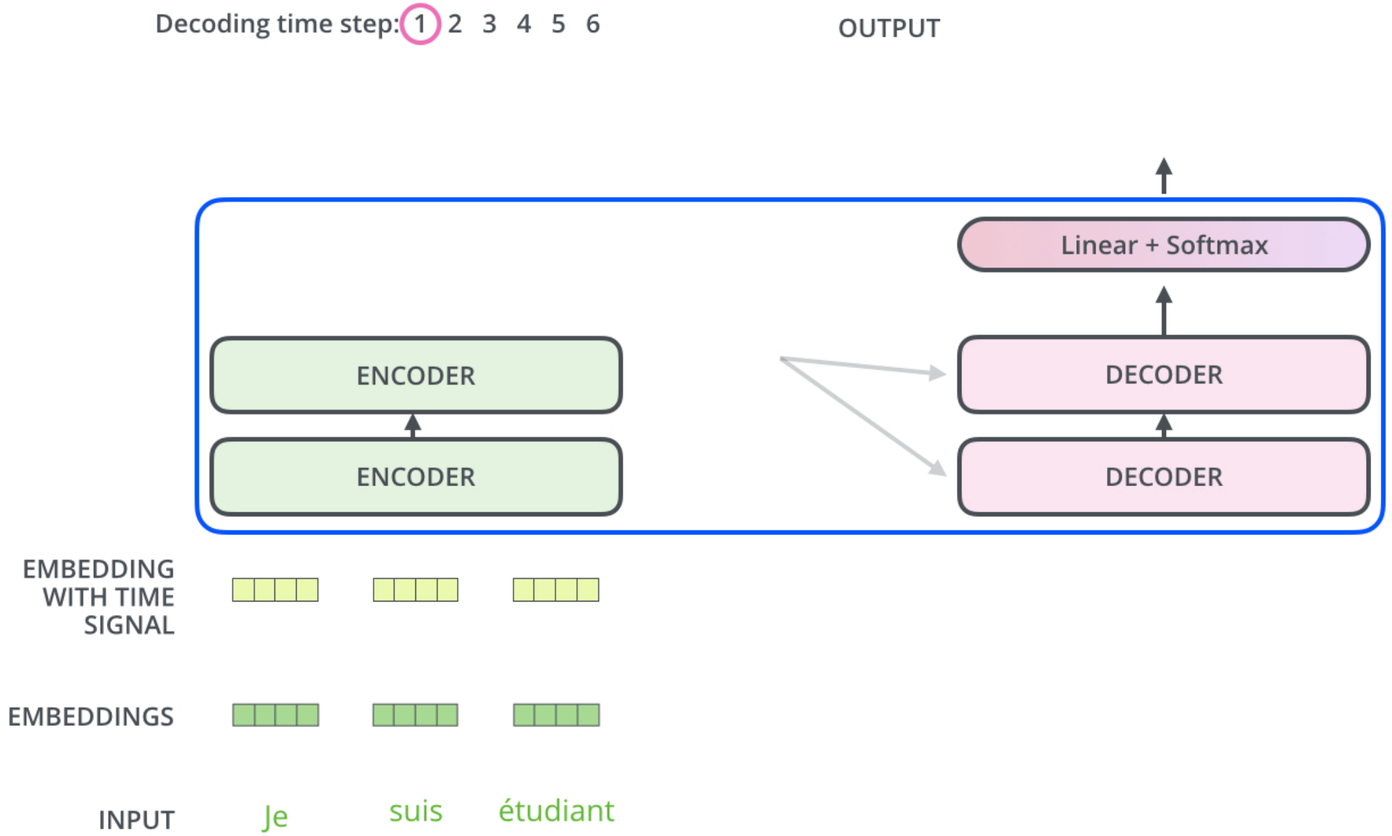


Attention is *all you need*



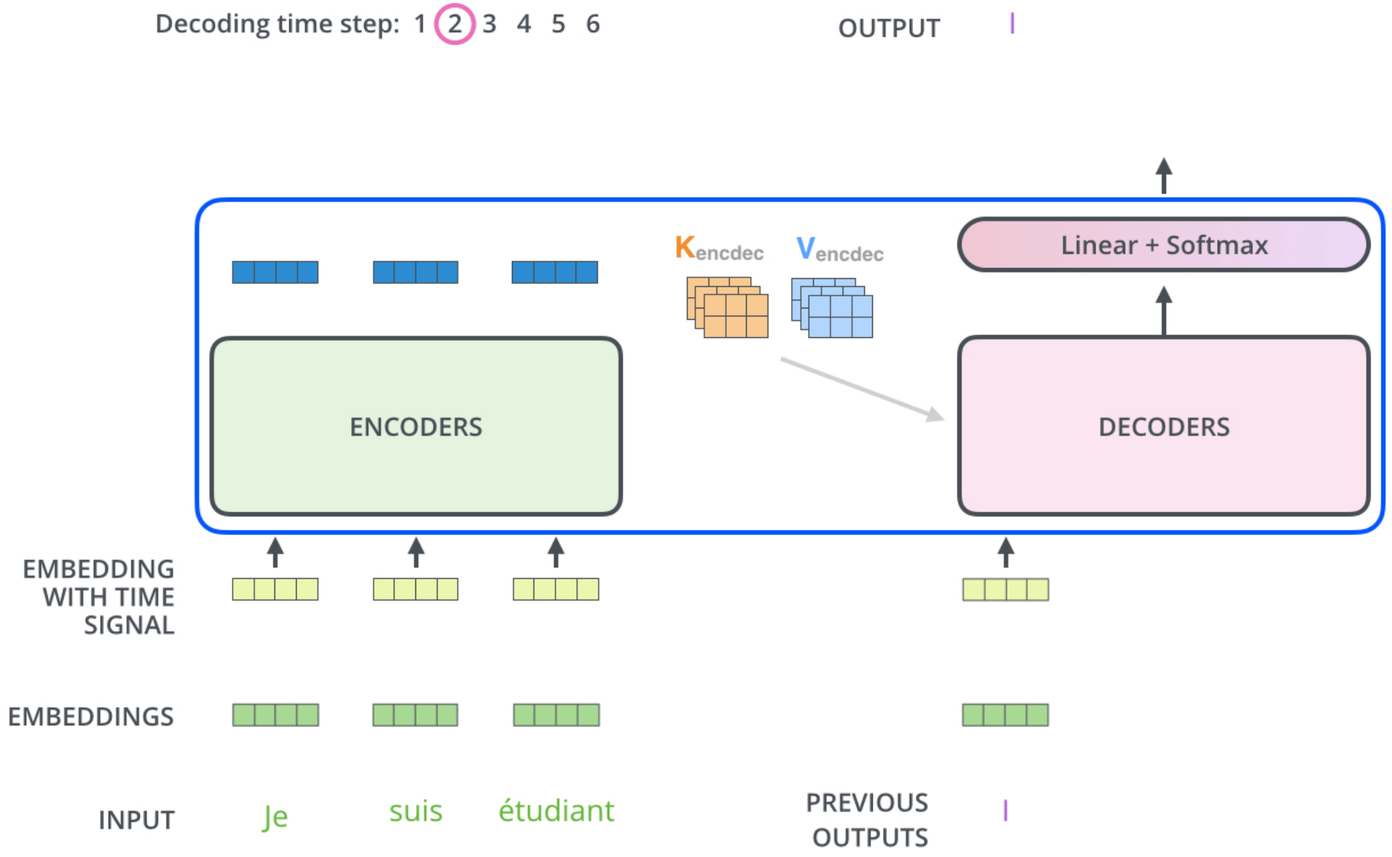
Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



TRANSFORMATEC

Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

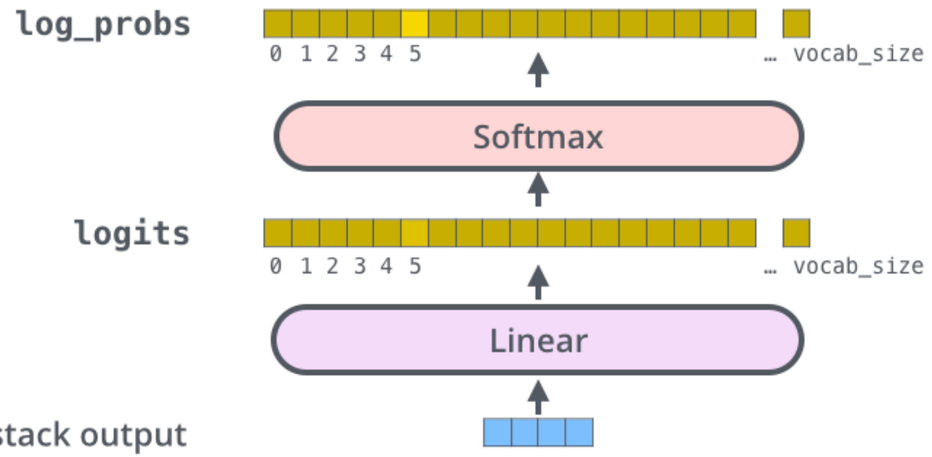
Attention is *all you need*

Which word in our vocabulary is associated with this index?

am

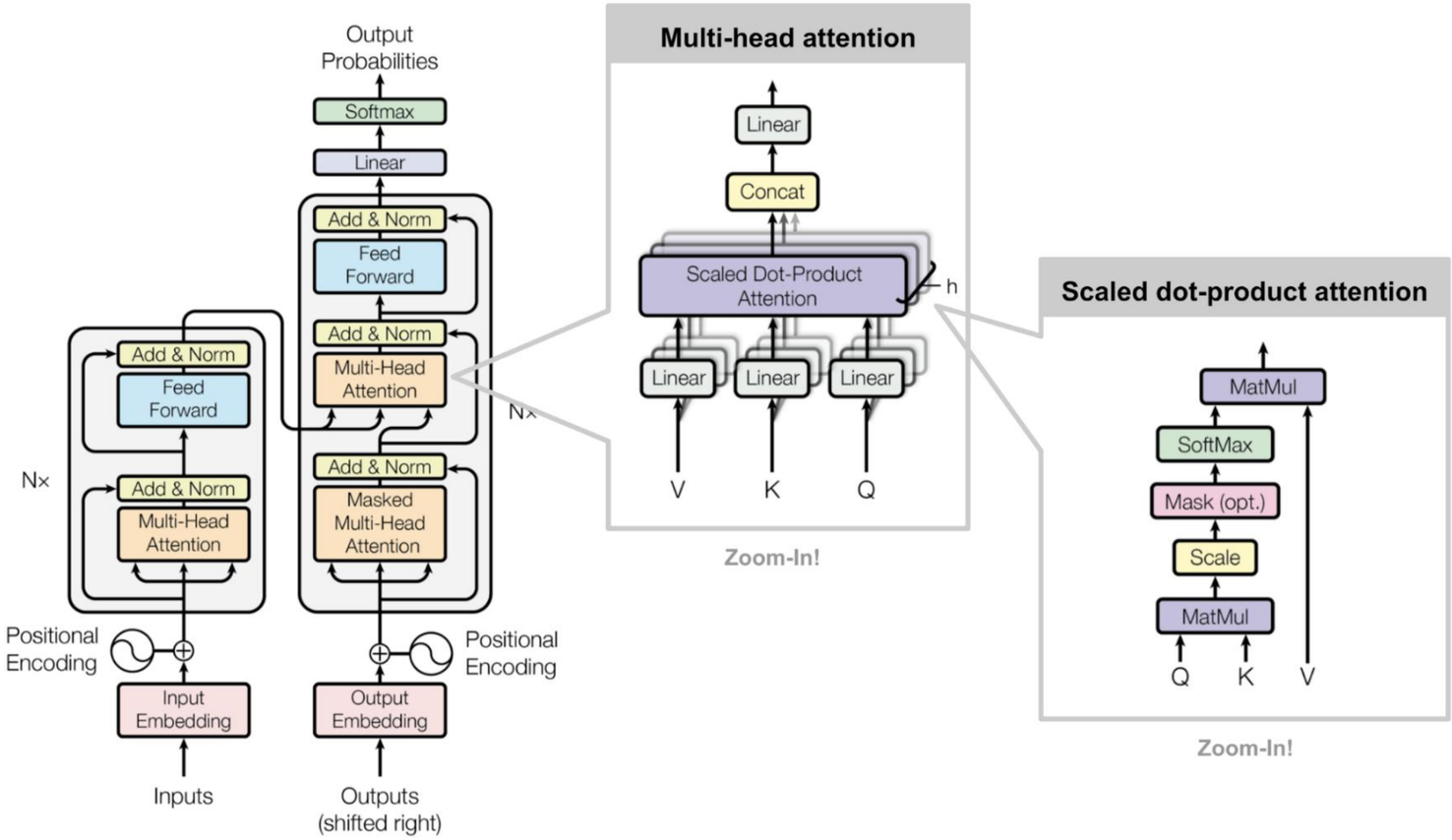
Get the index of the cell with the highest value (`argmax`)

5



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

Attention is *all you need*



Ashish Vaswani et al. (2017) "Attention is all you need".
31st Conference on Neural Information Processing Systems (NIPS 2017)

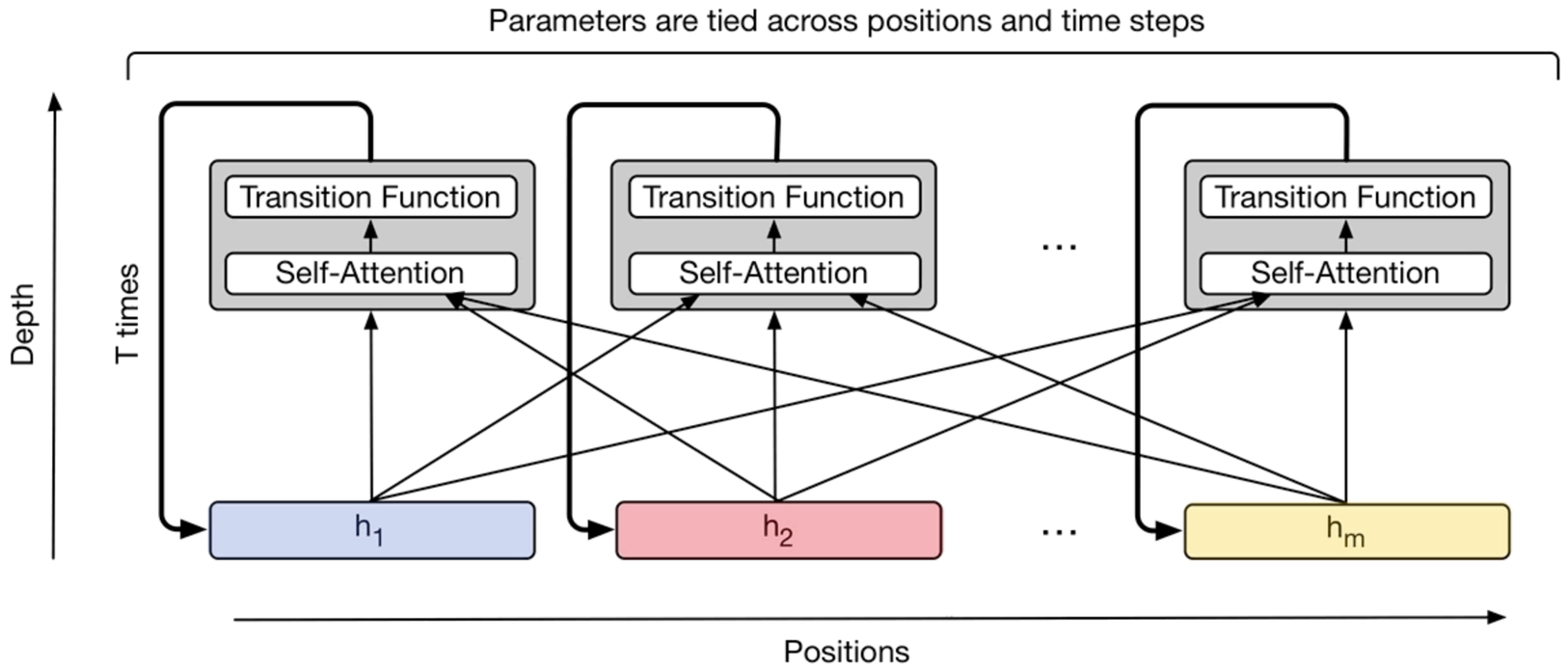
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3.

Universal *Transformers*

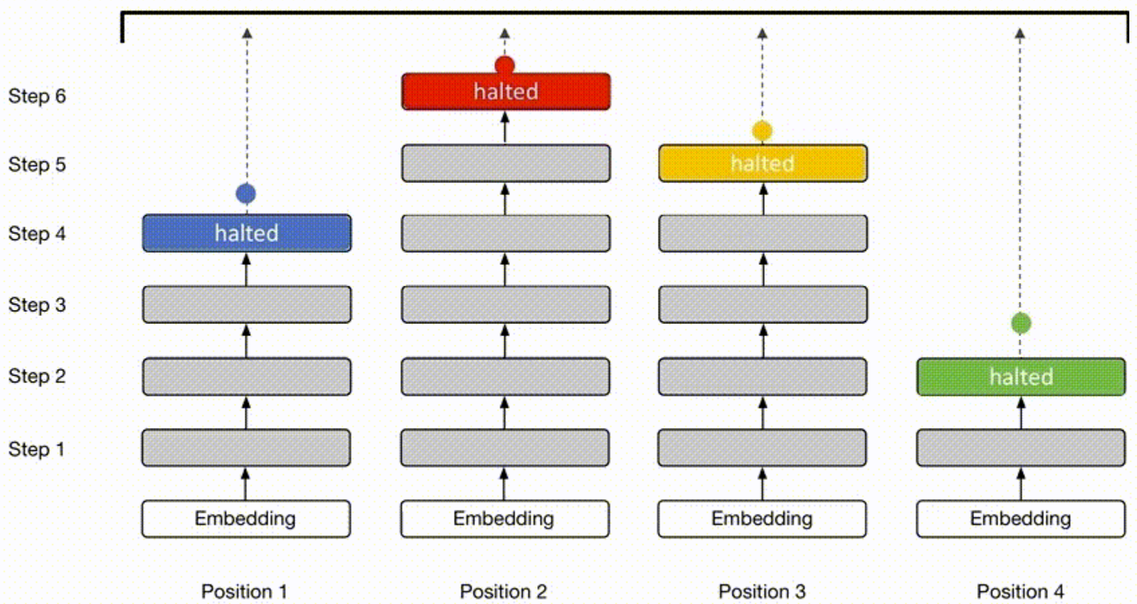
TRANSFORMATEC

Universal *Transformers*



Mostafa Dehghani et al. (2018) "Universal Transformers". International Conference on Learning Representations (ICLR).

Universal *Transformers*



Adaptive Computation Time

Mostafa Dehghani et al. (2018) "Universal Transformers". International Conference on Learning Representations (ICLR).

Universal *Transformers*

Adaptive Computation Time

Permite que el modelo determine de manera dinámica cuántas iteraciones son necesarias para procesar cada parte de un conjunto de datos.

En cada iteración t, el modelo calcula un valor de halting $h_t \in \mathbb{R}^L$ score para cada entrada:

$$h_t = \text{Sigmoid}(W x_t)$$

El progreso acumulado P_t

$$P_t = P_{t-1} + h_t$$

El modelo detiene el procesamiento de la entrada i cuando $\mathrm{P_t}[i] \geq 1$.

Alex Graves (2016) "Adaptive Computation Time for Recurrent Neural Networks". arXiv preprint arXiv:1603.08983.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

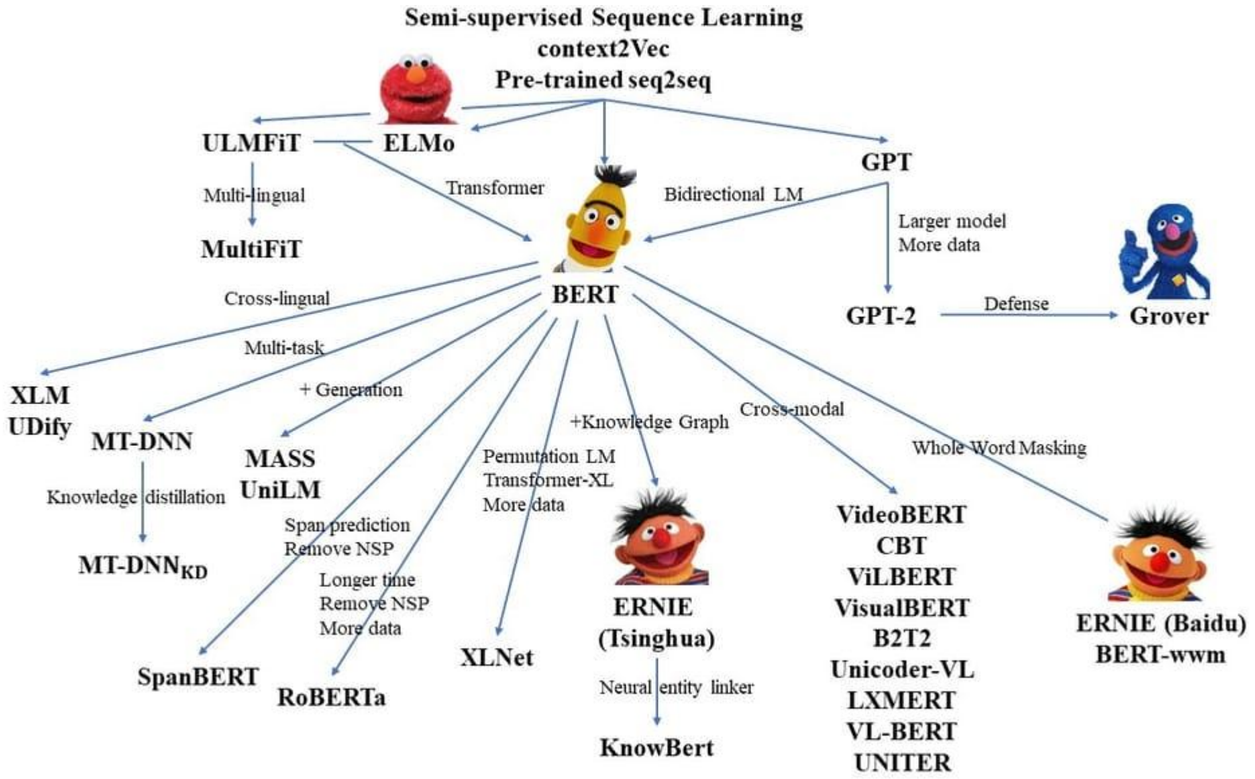
> Reinventa el mundo <

4.

BERT

TRANSFORMATEC

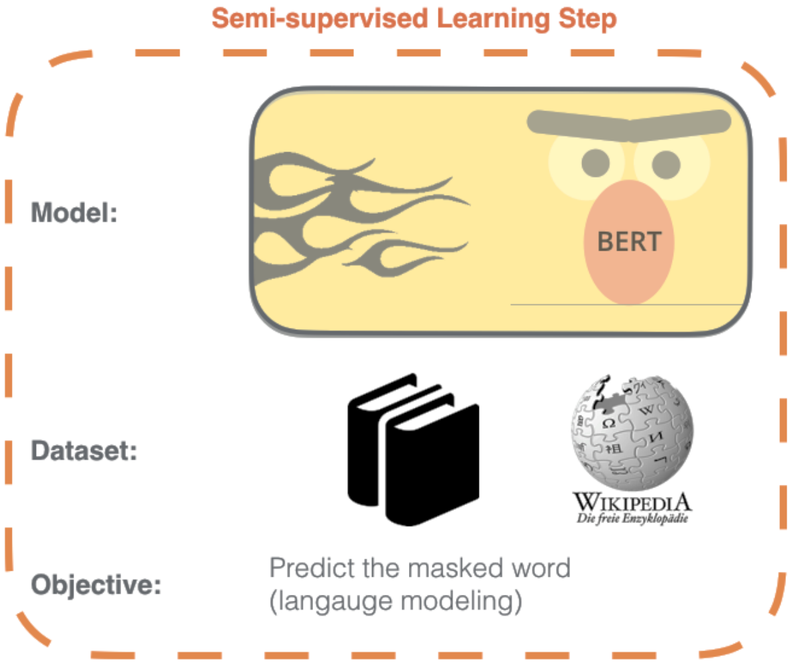
Sesame *Street*



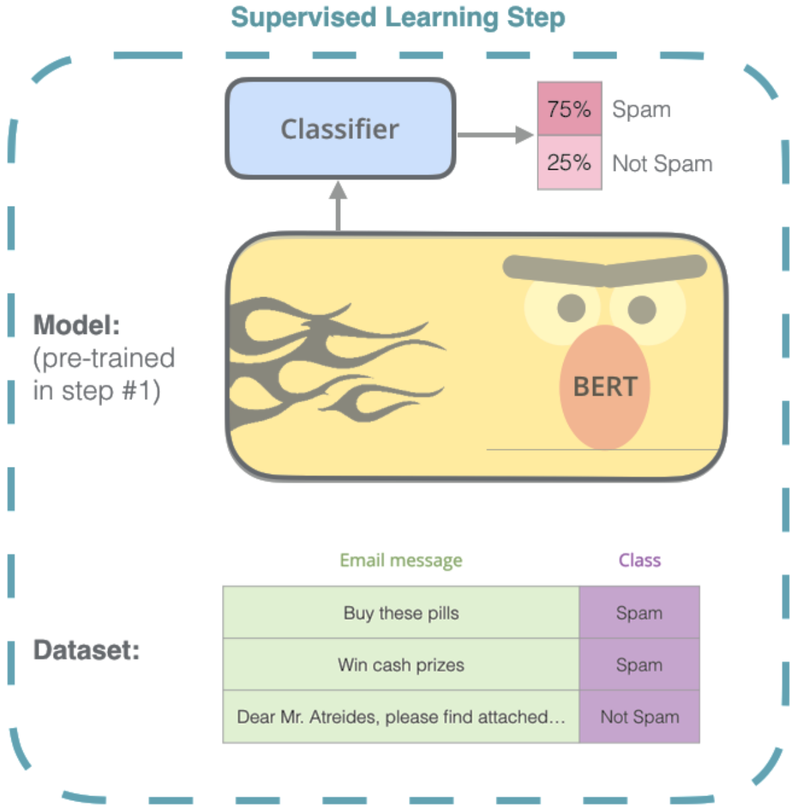
BERT

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



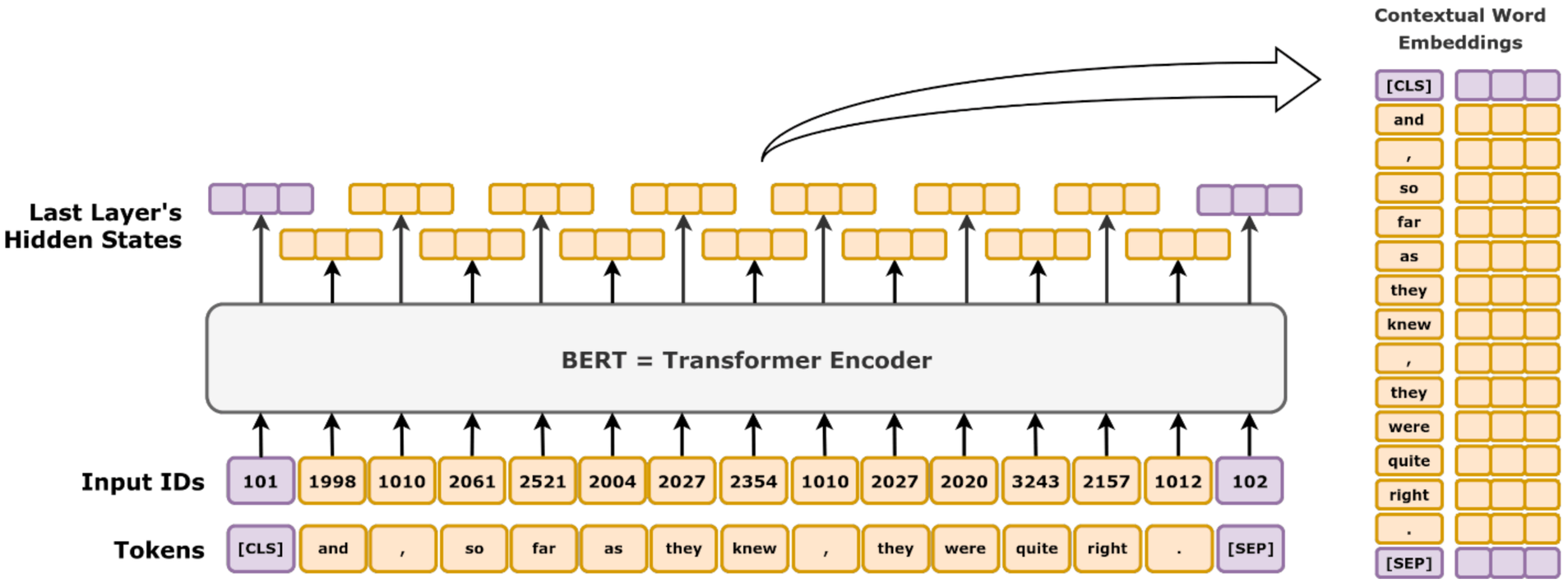
2 - Supervised training on a specific task with a labeled dataset.



TRANSFORMATEC

Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT



Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT

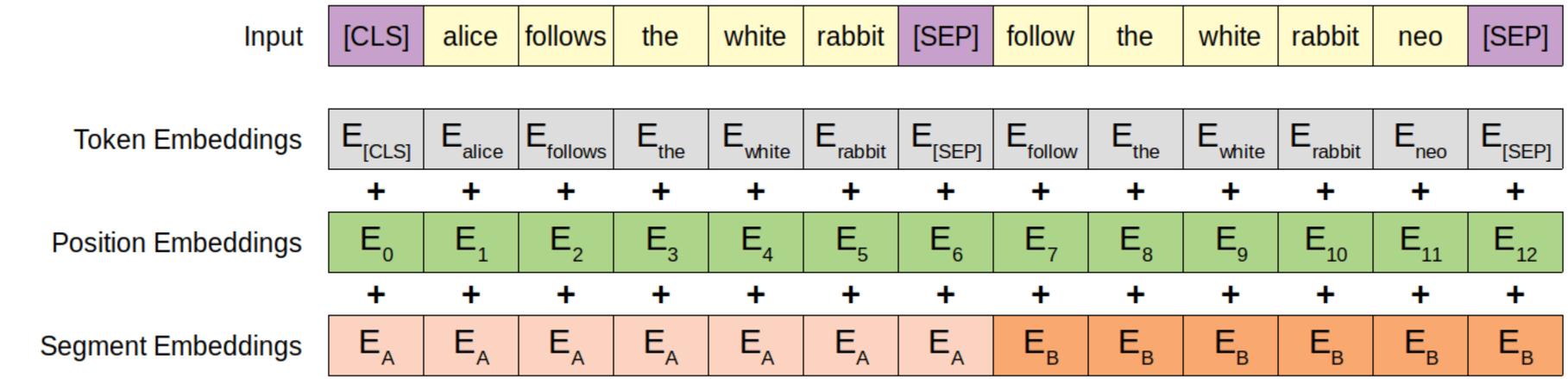
Input Text:

"WordPiece tokenization is a powerful technique in NLP."

WordPiece

Tokenization:

Word	Piece	token	ization	is
a	powerful	technique	in	N
L	P	.		

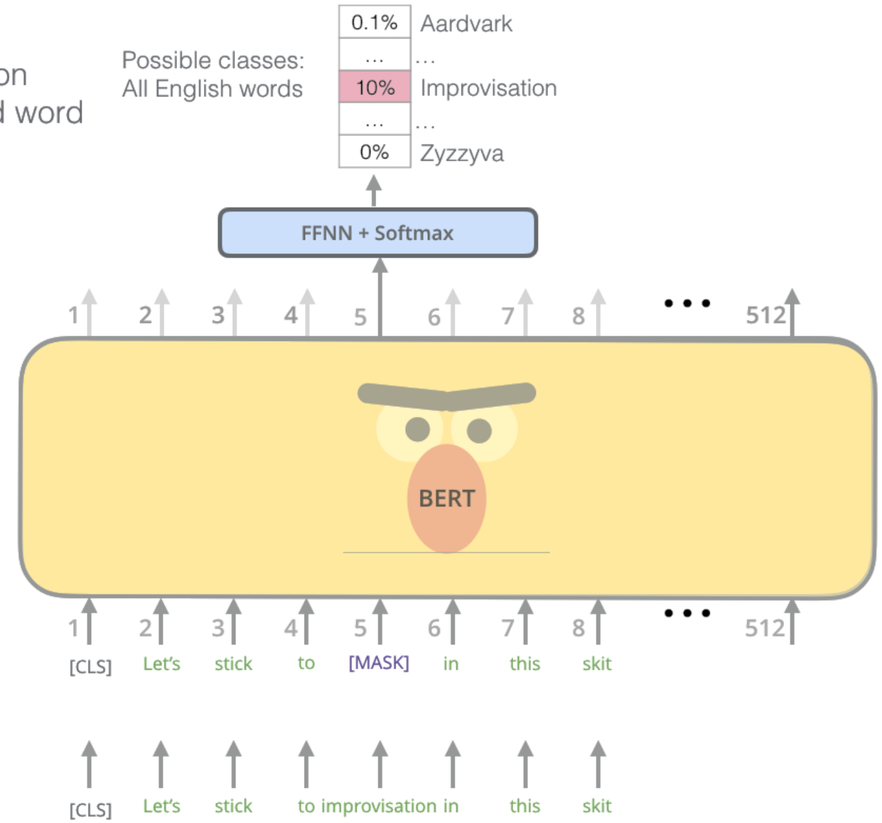


Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT: *Pre-training Tasks*

Masked Language Model

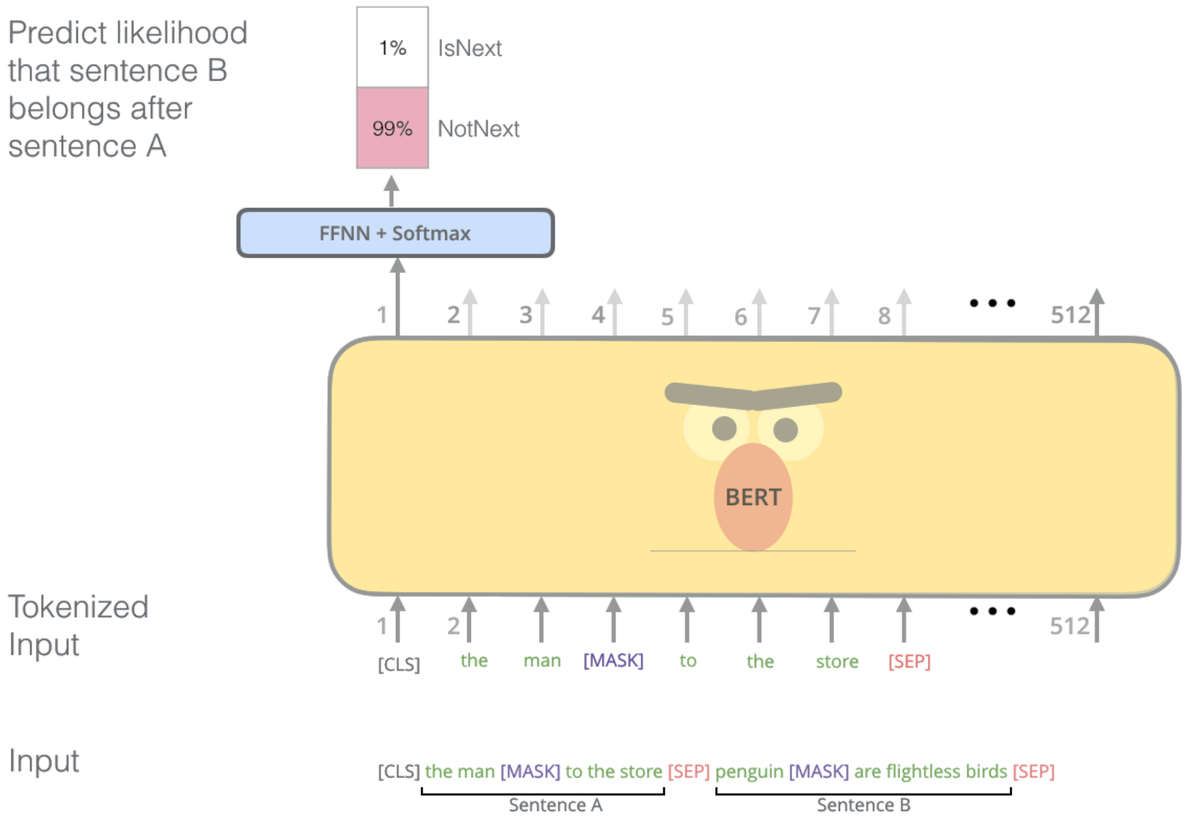
Use the output of the masked word's position to predict the masked word



Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT: *Pre-training Tasks*

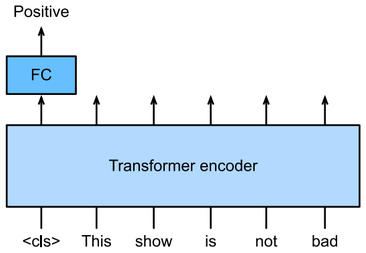
Two-sentence Tasks



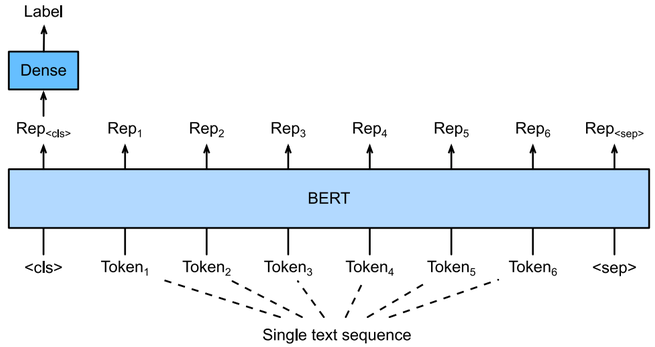
Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT: *Pre-training Tasks*

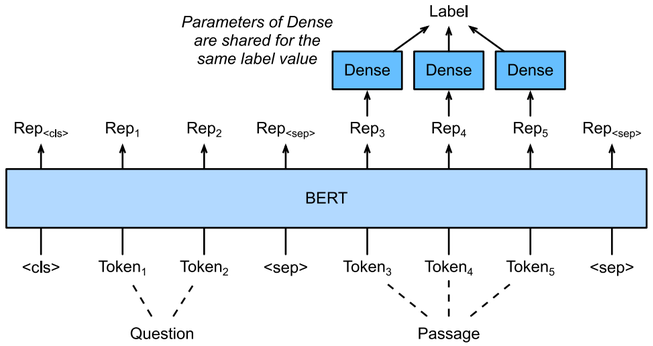
Sentiment classification



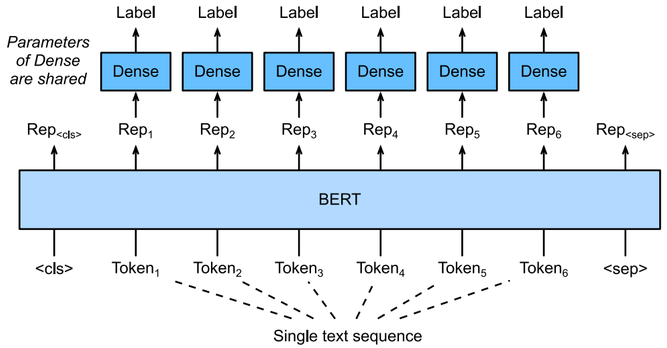
Sentence classification



Answering multiple-choice questions



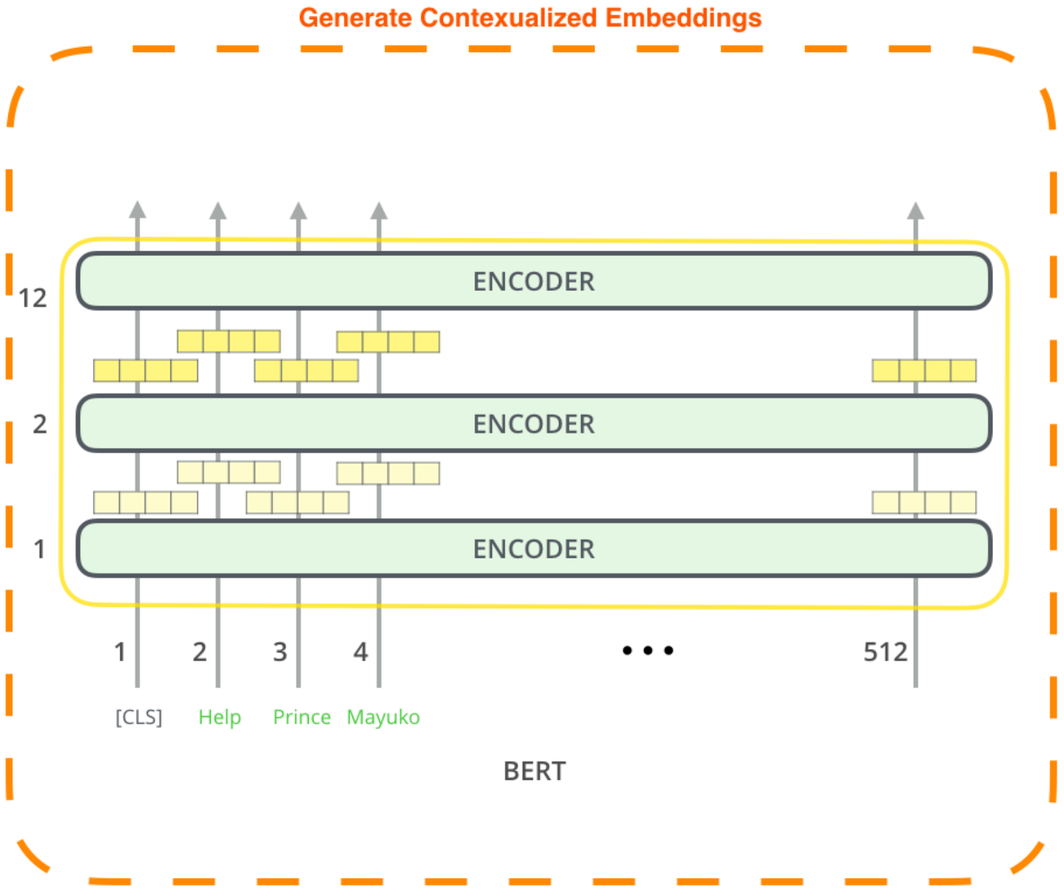
Part-of-speech tagging



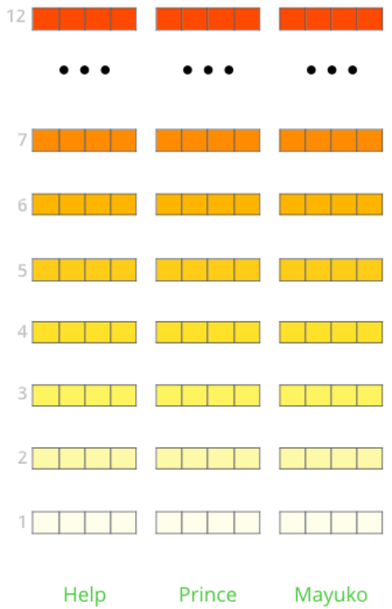
TRANSFORMATEC

Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT: *Feature extraction*



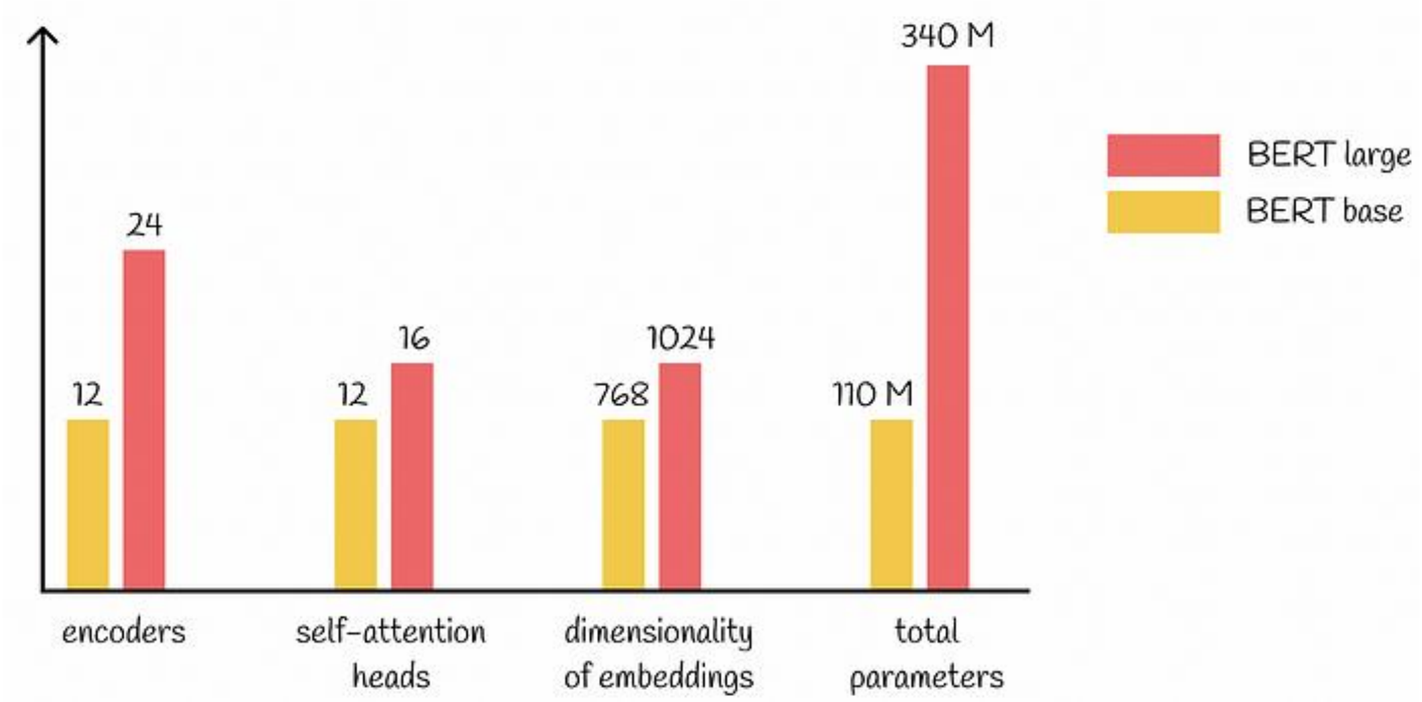
The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

BERT



Jacob Devlin et al. (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Proceedings of naacL-HLT (Vol. 1, p. 2).

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

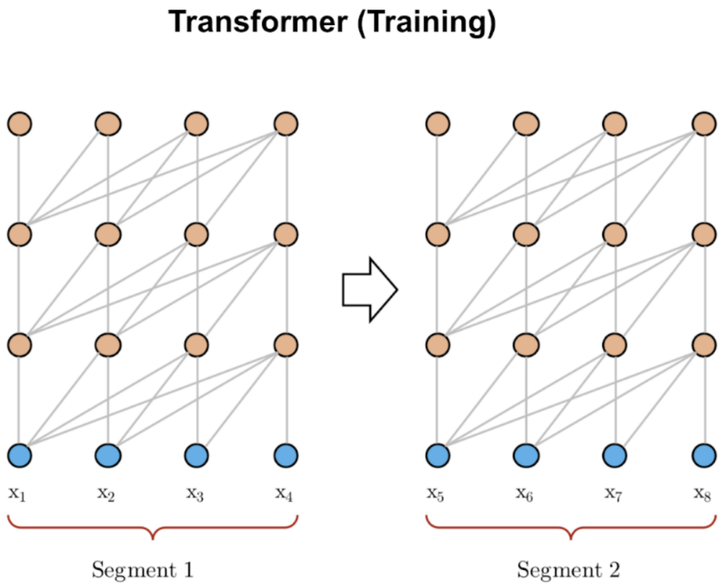
5.

Transformer-*XL*

TRANSFORMATEC

Transformer-*XL*

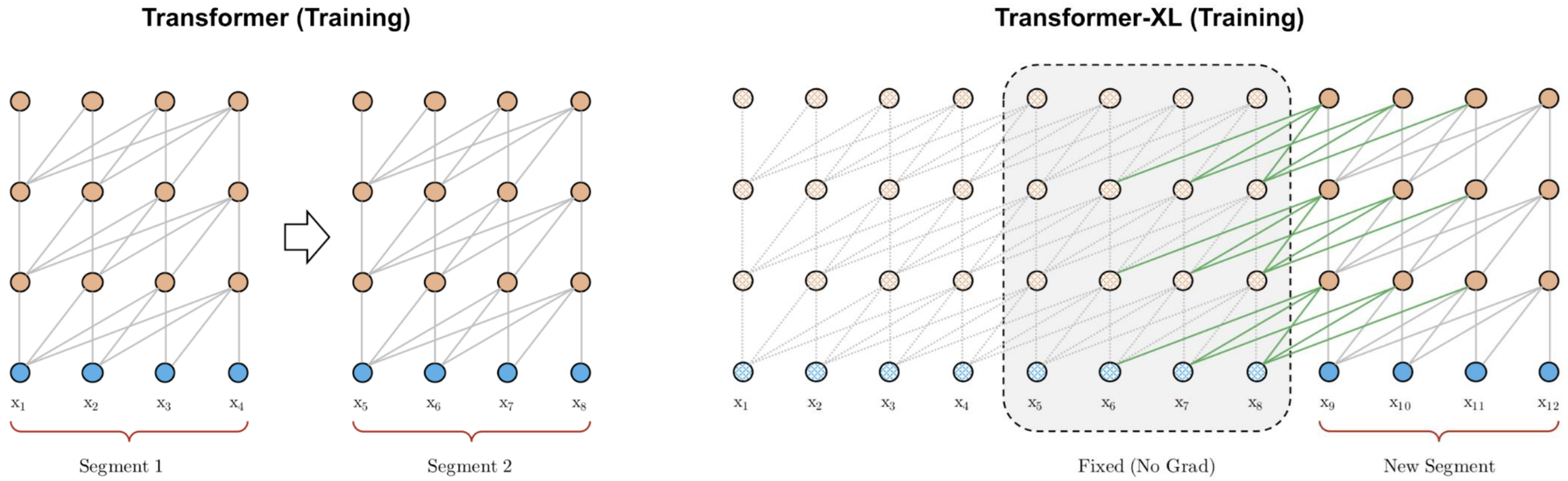
Ventana de contexto fija



Zihang Dai et al. (2018) "Transformer-XL: Language Modeling with Longer-Term Dependency". International Conference on Learning Representations (ICRL).

Transformer-*XL*

Segment-Level Recurrence



Zihang Dai et al. (2018) "Transformer-XL: Language Modeling with Longer-Term Dependency". International Conference on Learning Representations (ICRL).

Transformer-*XL*

Segment-Level Recurrence

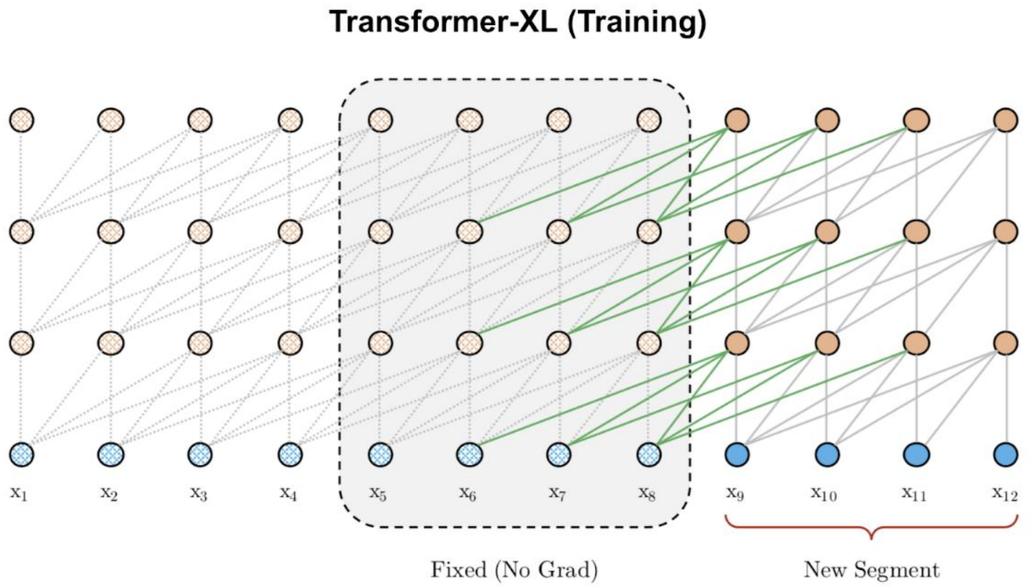
$$\tilde{h}_t^l = \left[\text{detach}\left(M_{t-1}^l\right), h_t^l\right]$$

Atención extendida

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}$$

K y V Generados tanto por el contexto recurrente (memoria) como el segmento actual

Q Generados por la secuencia actual



Zihang Dai et al. (2018) "Transformer-XL: Language Modeling with Longer-Term Dependency". International Conference on Learning Representations (ICRL).

Transformer-XL

Relative Positional Embedding (RPE)

$$A_{ij} = \frac{Q_i K_j^\top + Q_i r_{i-j}^\top + u K_j^\top + v r_{i-j}^\top}{\sqrt{d_k}}$$

- $Q_i r_{i-j}^\top$: Modela explícitamente la relación relativa de posición entre las entradas i (query) y j (key) mediante el embedding relativo r_{i-j}.
- $u K_j^\top$ y $v r_{i-j}^\top$: Es el sesgo global que capturan efectos posicionales independientes del query y key.
- r_{i-j}: Un embedding que representa la relación relativa de posición entre i y j.
- u, v: Vectores entrenables, que permiten al modelo ajustar los efectos globales de la posición relativa.

Zihang Dai et al. (2018) "Transformer-XL: Language Modeling with Longer-Term Dependency". International Conference on Learning Representations (ICRL).

Transformer-XL

Relative Positional Embedding (RPE)

$$A_{ij} = \frac{Q_i K_j^\top + Q_i r_{i-j}^\top + u K_j^\top + v r_{i-j}^\top}{\sqrt{d_k}}$$

Algunos autores omiten los dos últimos términos, asumiendo que las dependencias globales no son relevantes.

Esta aproximación puede ser valida en secuencias cortas.

Zihang Dai et al. (2018) "Transformer-XL: Language Modeling with Longer-Term Dependency". International Conference on Learning Representations (ICRL).

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

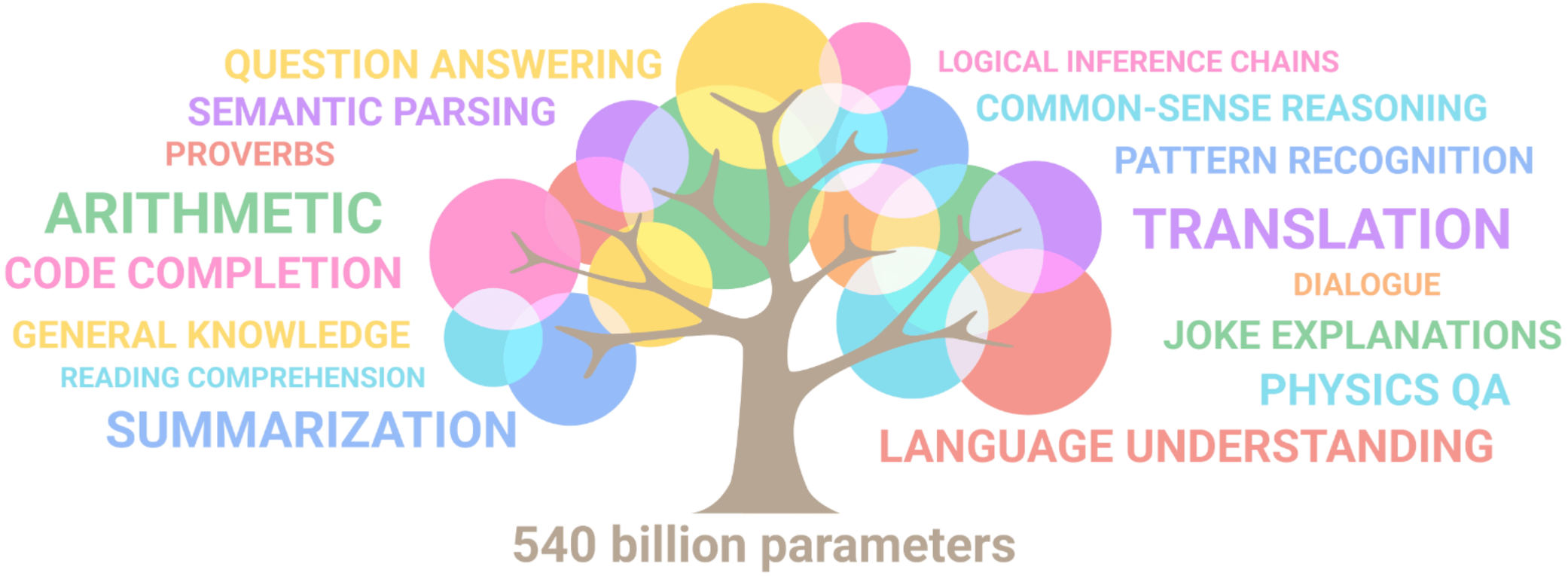
6.

PaLM

TRANSFORMATEC

PaLM

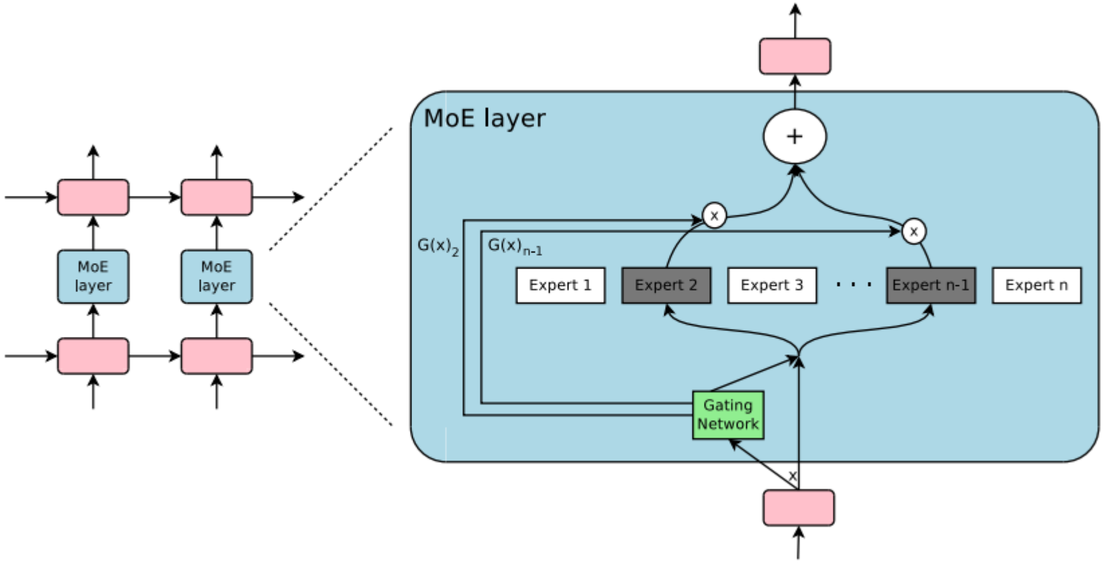
(Pathways Language Model)



Aakanksha Chowdhery et al. (2022) "PaLM: Scaling Language Modeling with Pathways". Journal of Machine Learning Research, 24(240), 1-113.

PaLM
(Pathways Language Model)

Mixture-of-Experts (MoE)

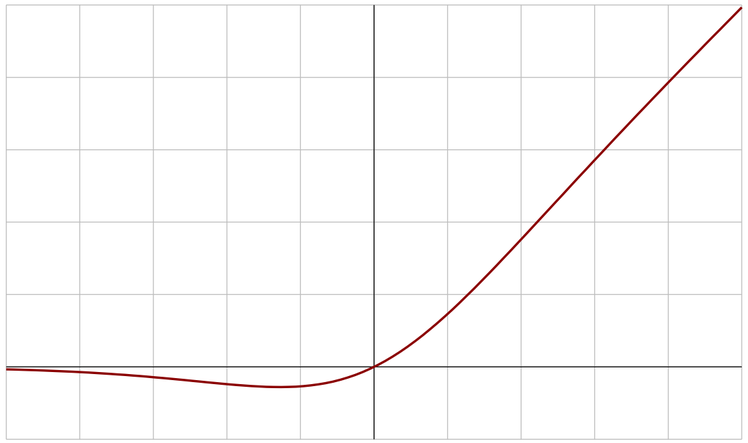


Aakanksha Chowdhery et al. (2022) "PaLM: Scaling Language Modeling with Pathways". Journal of Machine Learning Research, 24(240), 1-113.

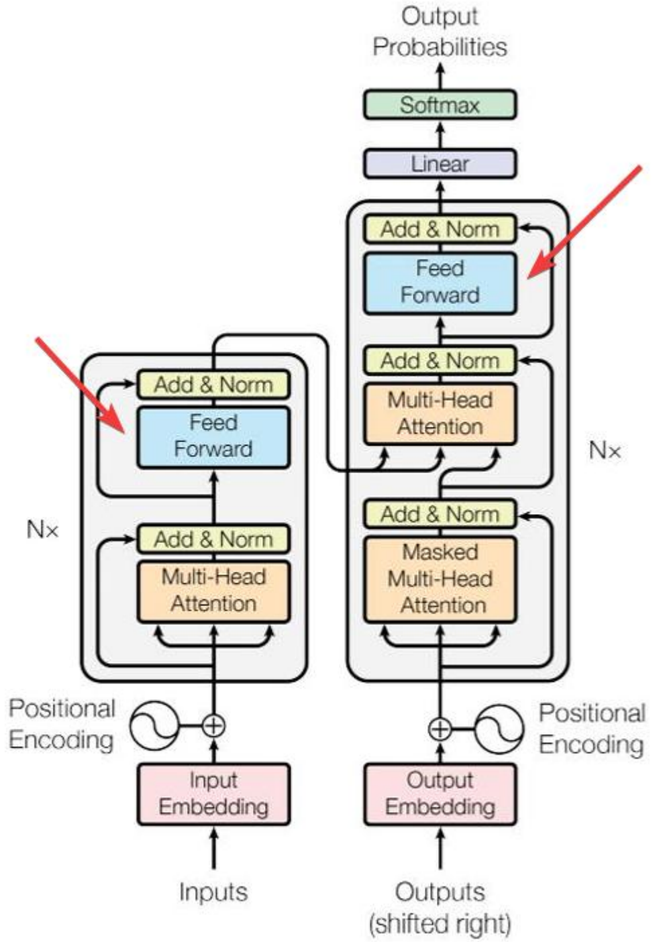
PaLM
(Pathways Language Model)

SwiGLU activation

$$\text{Swish}(xW_1) \cdot xW_2$$



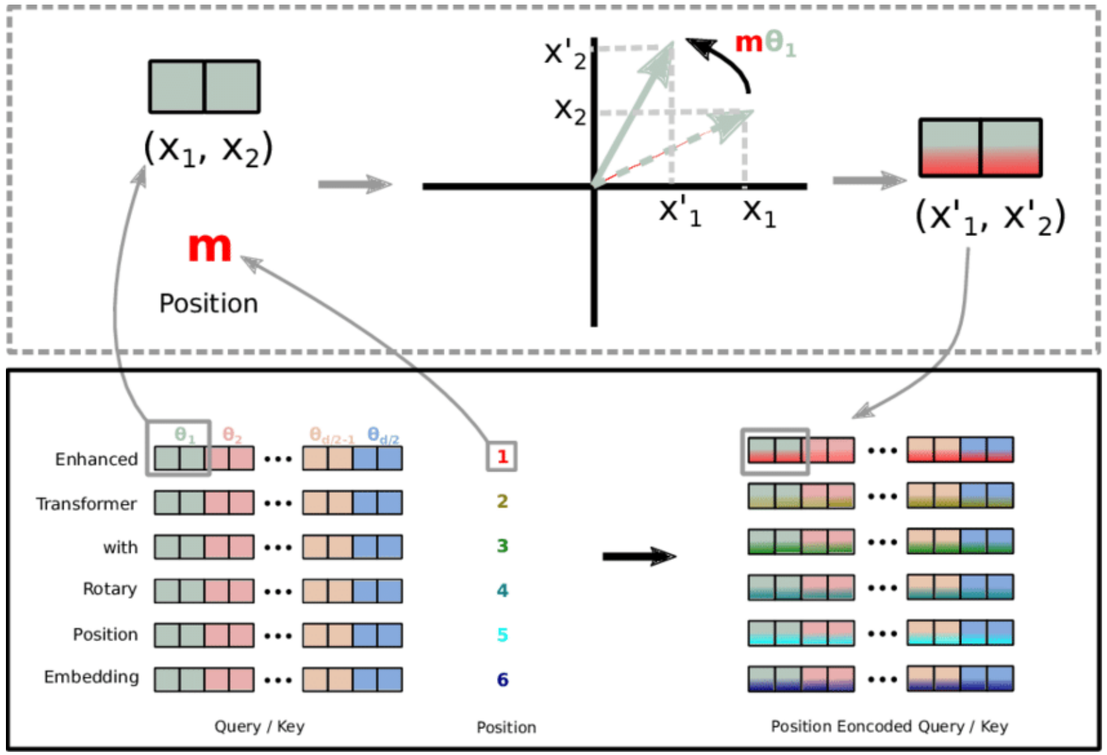
$$\text{Swish}(x) = x\,\text{Sigmoid}(x)$$



Aakanksha Chowdhery et al. (2022) "PaLM: Scaling Language Modeling with Pathways". Journal of Machine Learning Research, 24(240), 1-113.

PaLM
(Pathways Language Model)

RoPE (Rotary Positional Embeddings)



Jianlin Su et al. (2021) "RoFormer: Enhanced Transformer with Rotary Position Embedding". Neurocomputing, 568, 127063.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

GRACIAS

Victor Flores Benites

TRANSFORMATEC