Procesamiento de Lenguaje Natural Avanzado

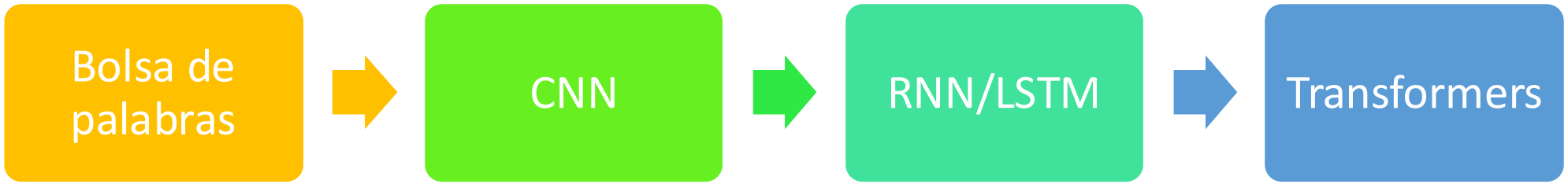
Atención - Transformers

Dra. Helena Gómez Adorno
helena.gomez@iimas.unam.mx

Dr. Fazlourrahman Balouchzahi
fbalouc@iimas.unam.mx

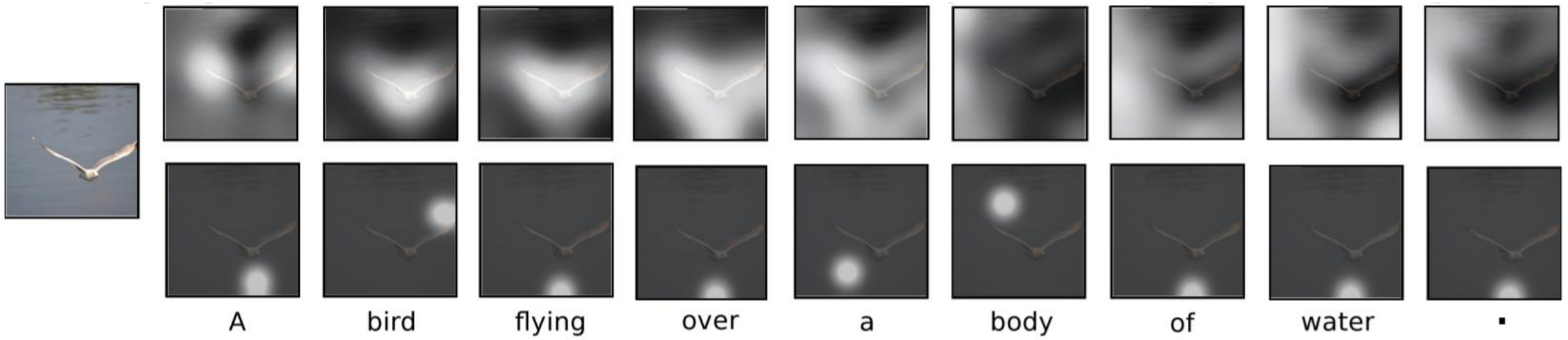
Correo del curso:
pln.cienciadedatos@gmail.com

Introducción



Atención

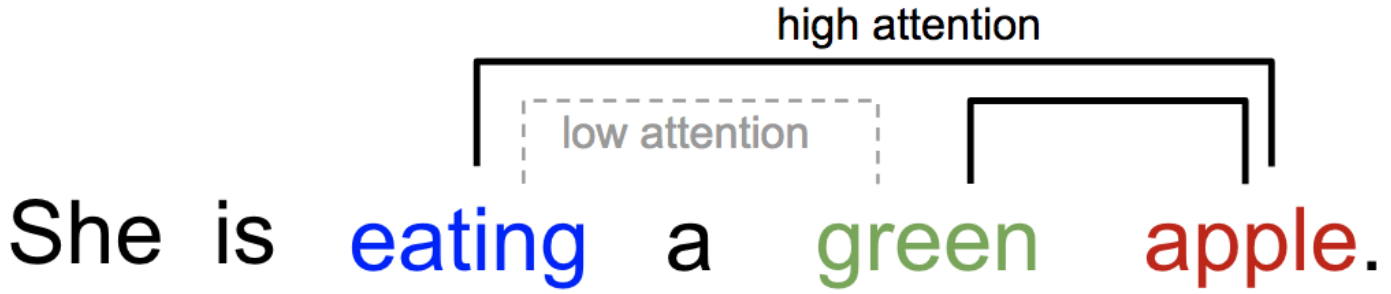
Los mecanismos de atención en redes neuronales se basan en la intuición de cómo los humanos perciben las imágenes y cómo se interpreta una secuencia de palabras.



Atención

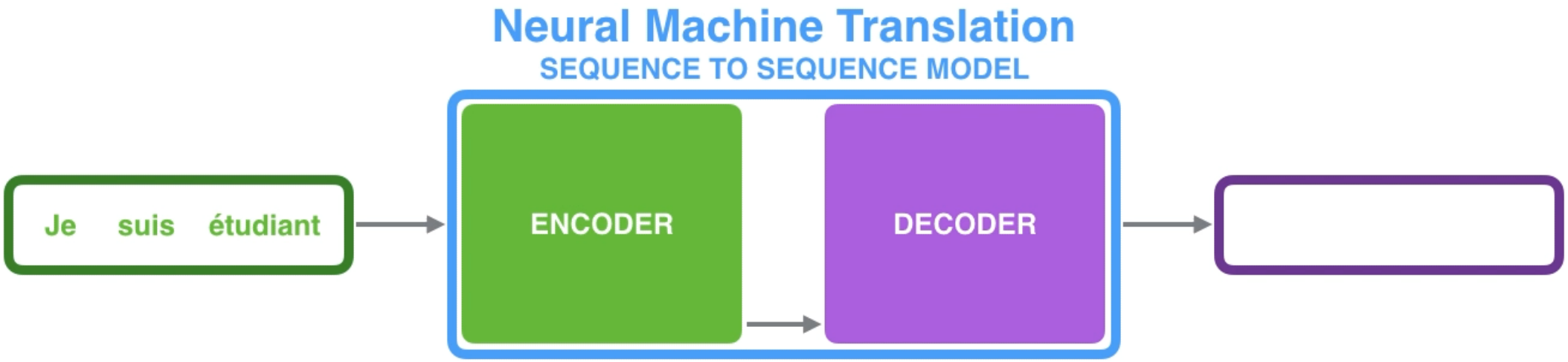
Los mecanismos de atención en redes neuronales se basan en la intuición de cómo los humanos perciben las imágenes y cómo se interpreta una secuencia de palabras.

La atención consiste en definir un vector de pesos que permita ponderar los diferentes niveles de las entradas.



Seq2seq: Arquitectura Codificador-Decodificador

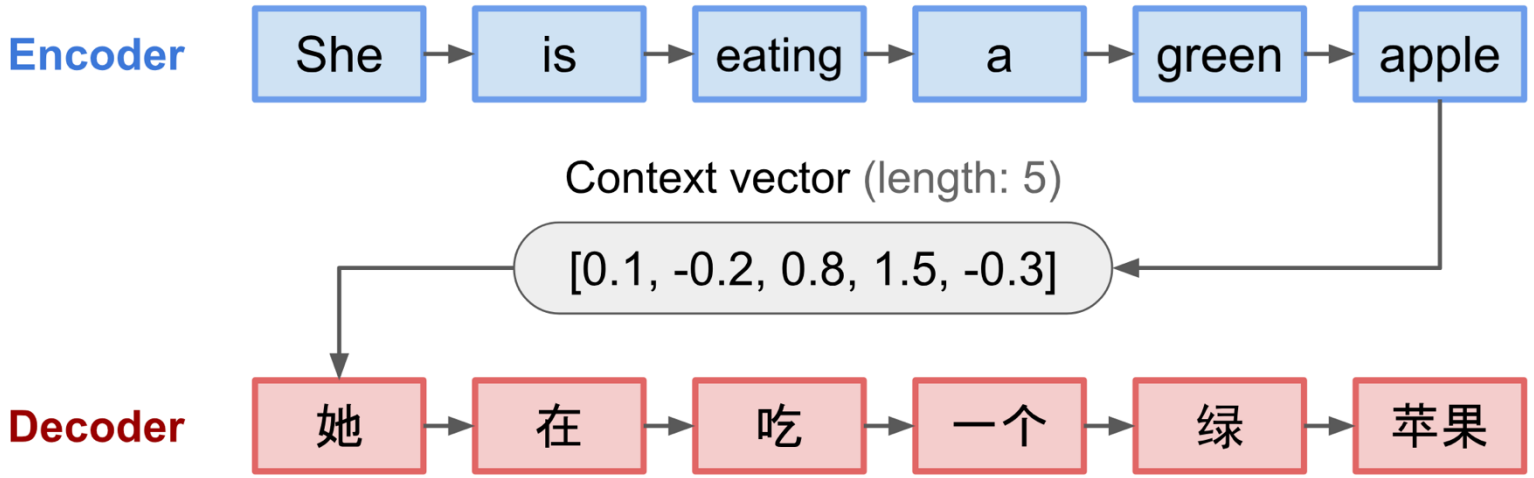
Son modelos que reciben una secuencia de entrada y producen una secuencia de salida. Ambas secuencias pueden ser de longitudes arbitrarias y no necesariamente de la misma magnitud. Este tipo de arquitectura se utiliza para problemas como traducción automática, sistemas pregunta respuesta y resumen automático.



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

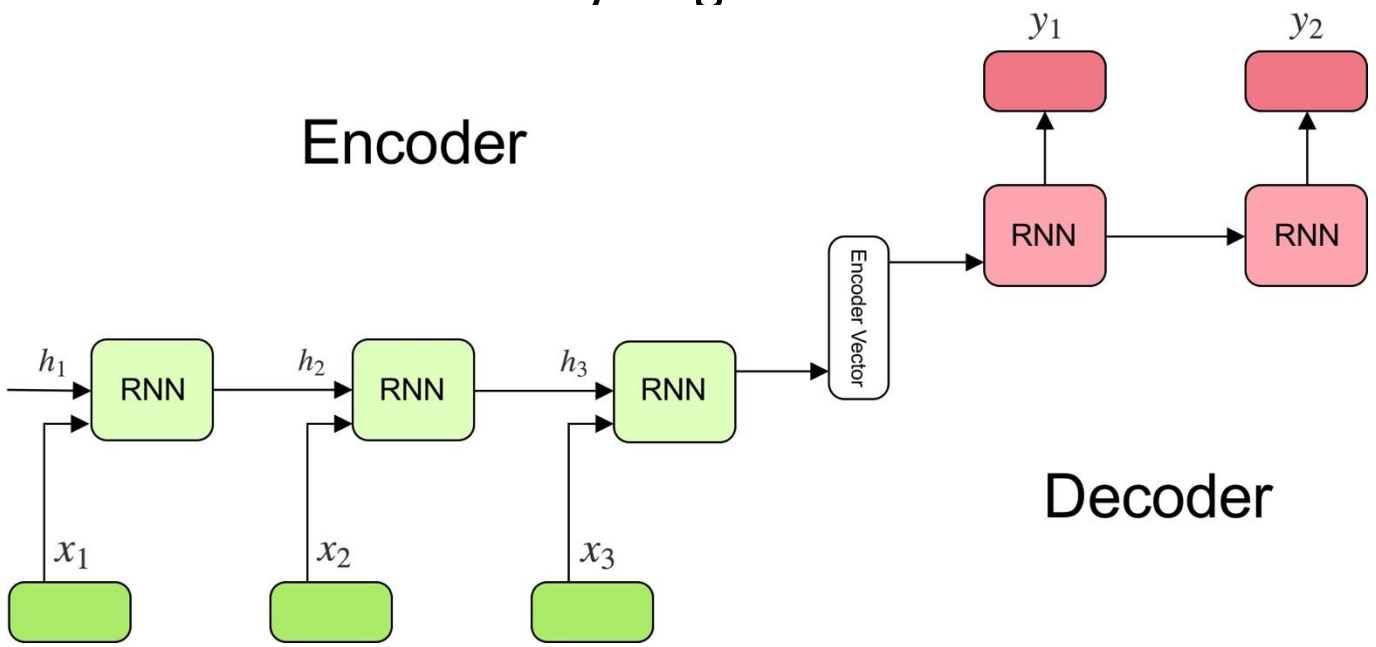
Arquitectura Codificador-Decodificador

Tras procesar toda la secuencia de entrada, el **codificador** envía el contexto al **descodificador**, que comienza a producir la secuencia de salida elemento por elemento. El vector de contexto es básicamente el número de unidades ocultas en la RNN codificadora.

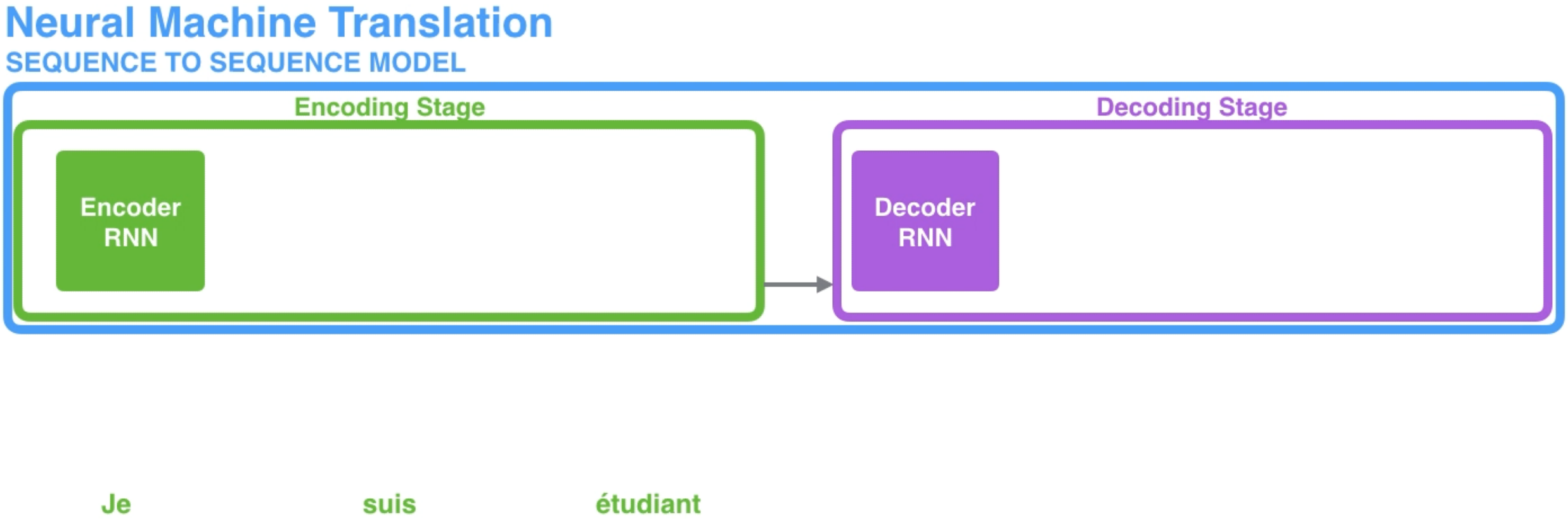


Arquitectura Codificador-Decodificador

- Permiten modelar relaciones entre elementos de una secuencia.
- Tanto el codificador como el decodificador se pueden implementar como una RNN.
- Se genera un vector contexto que representa la secuencia de entrada.
- Se complica que el vector contexto capture todas las propiedades de secuencias de entrada muy largas.



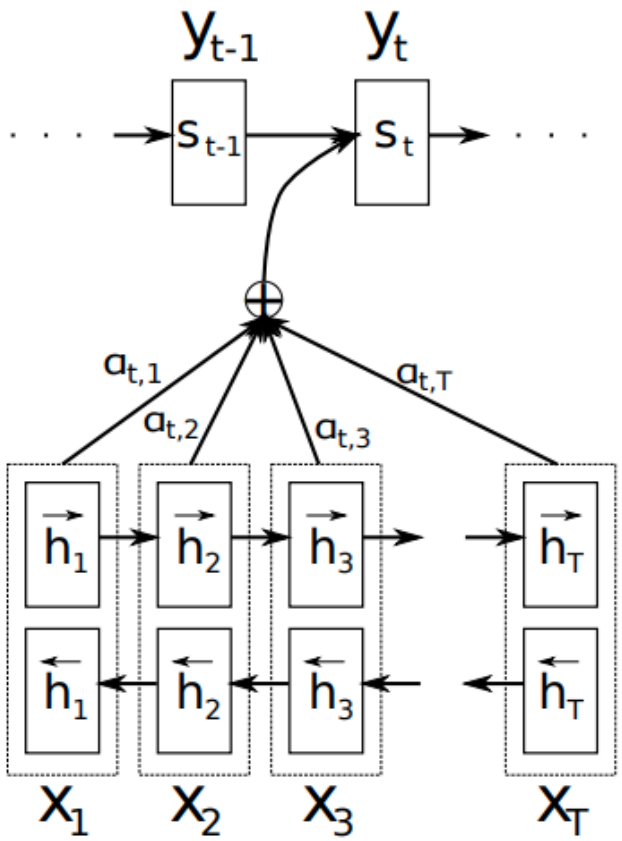
Arquitectura Codificador-Decodificador



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

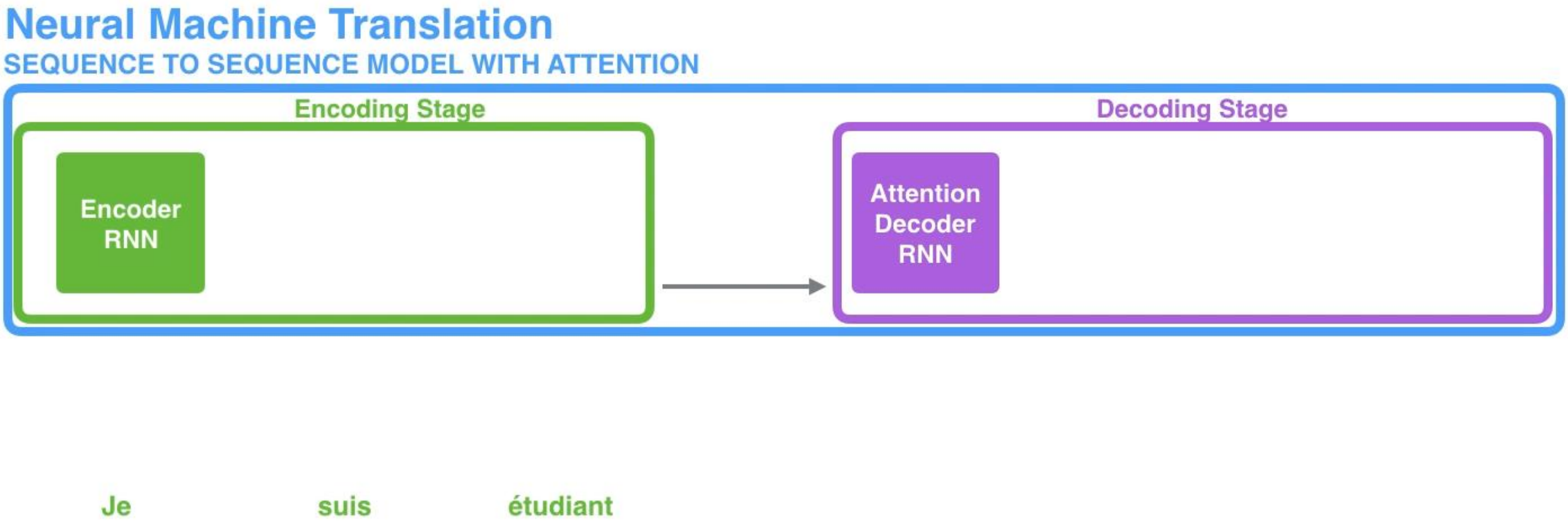
Codificador-Decodificador - Atención

Bahdanau 2015, propuso utilizar todos los estados ocultos del codificador para crear un contexto customizado para cada elemento de la secuencia de salida.



Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate.” ICLR 2015.

Ahora pongamos atención



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Ahora pongamos atención

https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Codificador-Decodificador - Atención

Estado oculto del codificador $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}], i = 1, \dots, n$

Contexto para la salida y_i $c_t = \sum_{i=1}^{n} a_{ti} h_i$

Score de que tan relacionadas están (y_t , x_i) $a_{ti} = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{i=1}^{n} \exp(score(s_{t-1}, h_{i,}))}$

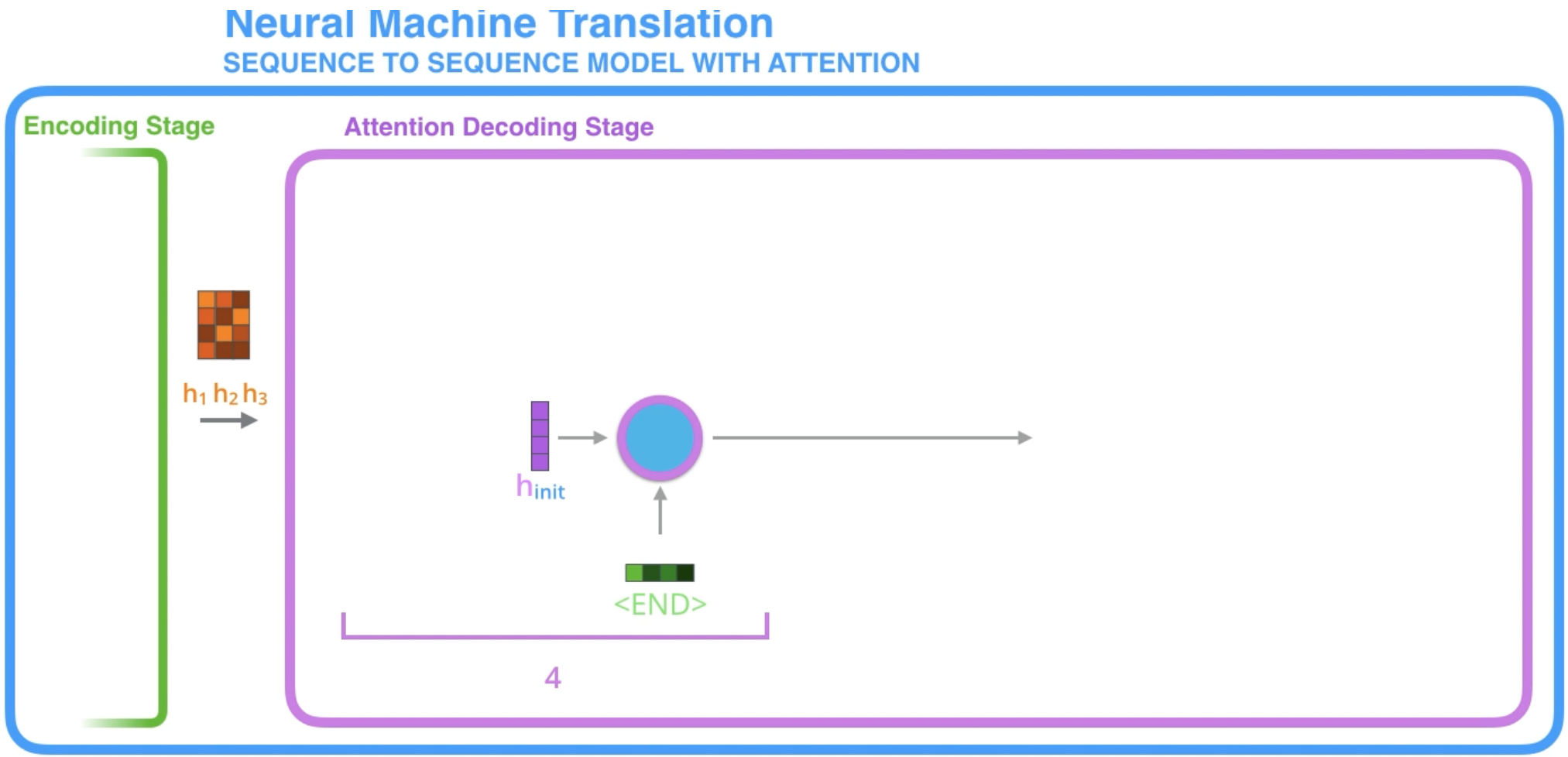
$$score(s_t, h_i) = v_a \tanh(W_a[s_t; h_i])$$

Estado oculto del decodificador $s_t = f(s_{t-1}, y_{t-1}, c_t), \qquad t = 1, \dots, m$

Atención – Funciones de Score

	Score	Artículo
Basado en Contenido	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves, 2014
Sumativo	$score(s_t, h_i) = v_a \tanh(W_a[s_t;\ h_i])$	Bahdanau, 2015
Basado en Localización	$\text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$	Luong, 2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t \mathbf{W}_a \boldsymbol{h}_i$	Luong, 2015
Producto Punto	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t \boldsymbol{h}_i$	Luong, 2015
Producto Punto Escalado	$score(s_t, h_i) = \frac{s_t h_i}{\sqrt{n}}$	Vaswani, 2017

Modelo Seq2seq con atención



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Atención – Variantes

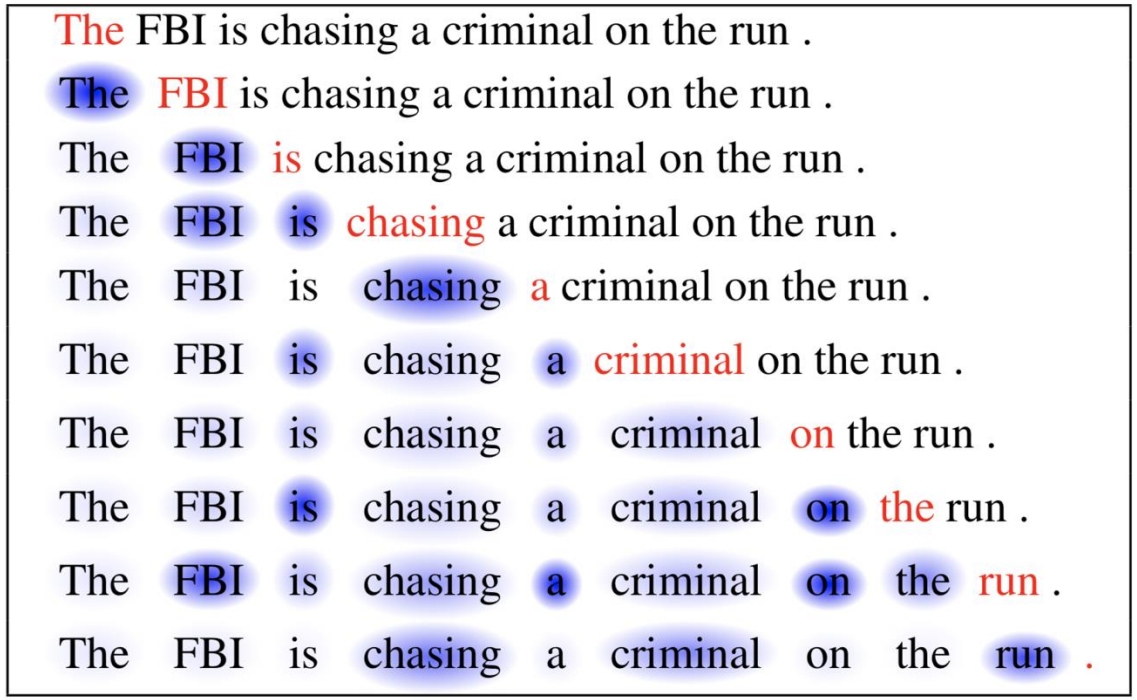
- **Global.** Se atiende a la secuencia completa.
- **Local.** Se atiende a una porción de la secuencia.
- **Auto-Atención.** Evalúa los pesos considerando la misma secuencia.

Para todas estas variantes se puede utilizar cualquier función de **score**, lo que varía es a que partes de la secuencia se tiene acceso.

Auto-Atención

Este mecanismo consiste en calcular un score de atención tomando en cuenta un elemento de una secuencia y evaluándolo contra otro elemento de la misma secuencia.

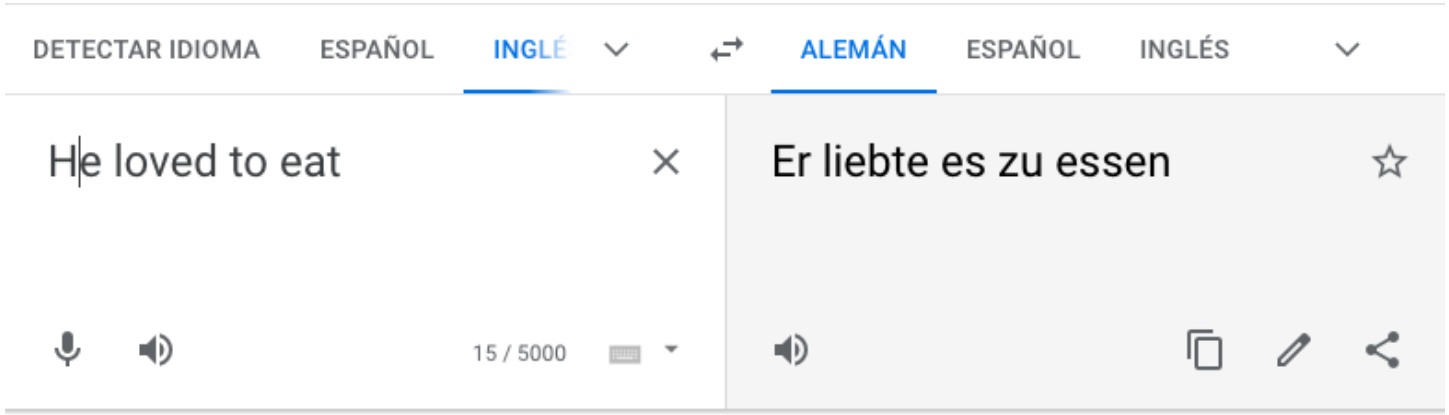
En el caso del texto este mecanismo permite evaluar las relaciones que existen entre palabras de la misma secuencia, no sólo entre la secuencia de entrada y salida.

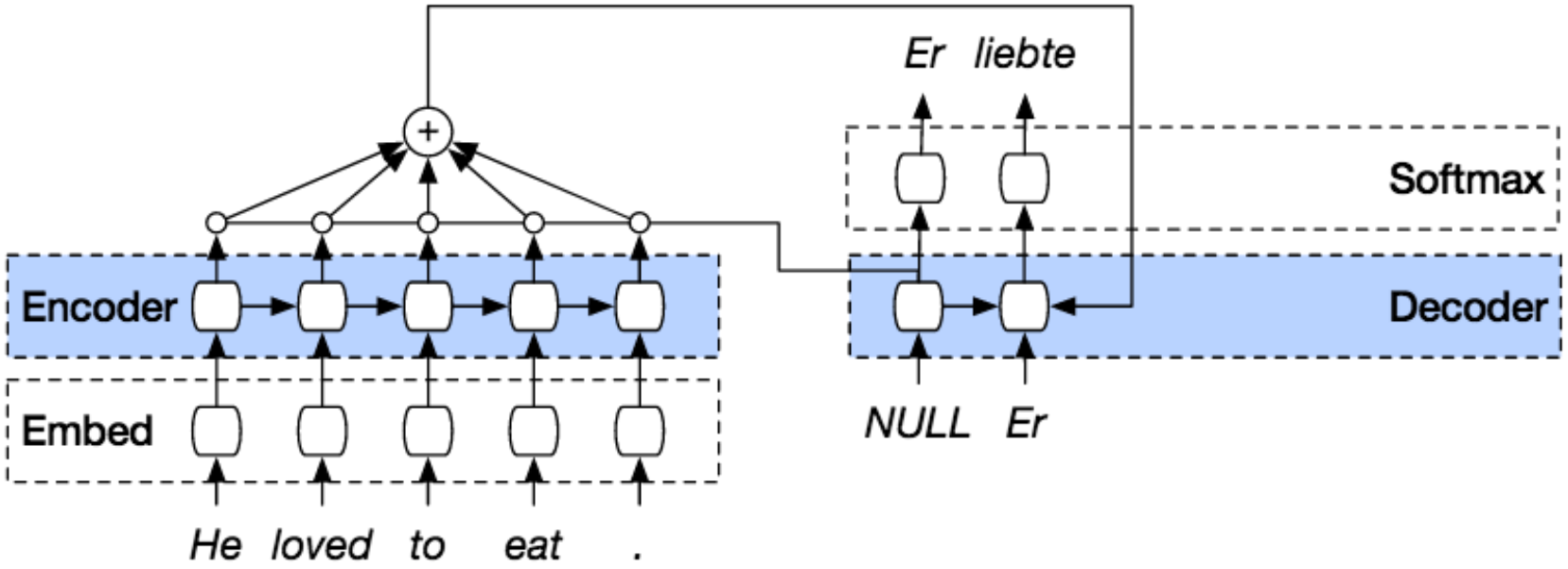


Jianpeng Cheng, Li Dong and Mirella Lapata, "Long Short-Term Memory-Networks for Machine Reading", 2016

Ejemplo – Traducción Automática

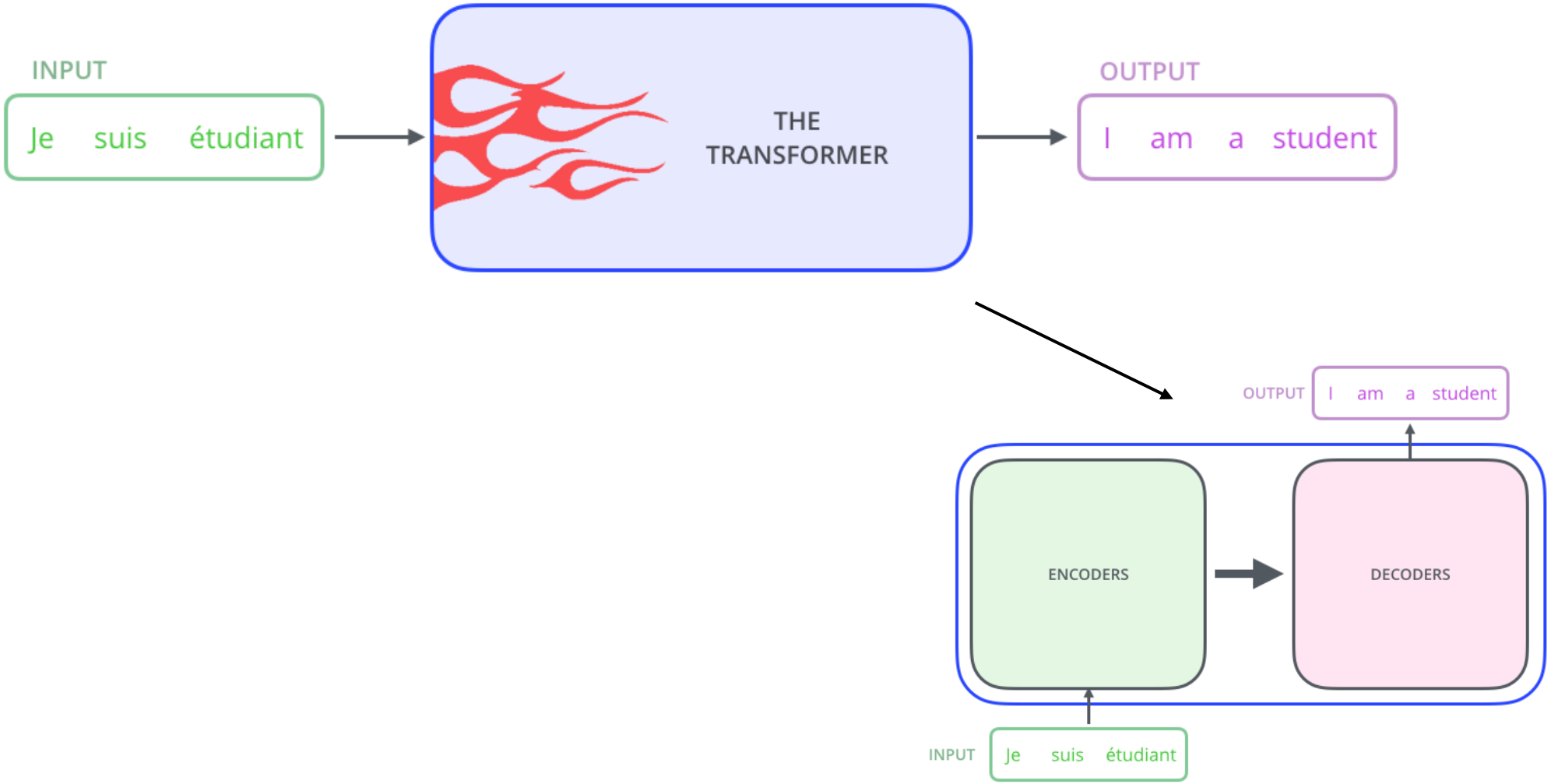
Tarea de traducir un texto de un lenguaje a otro utilizando una computadora.





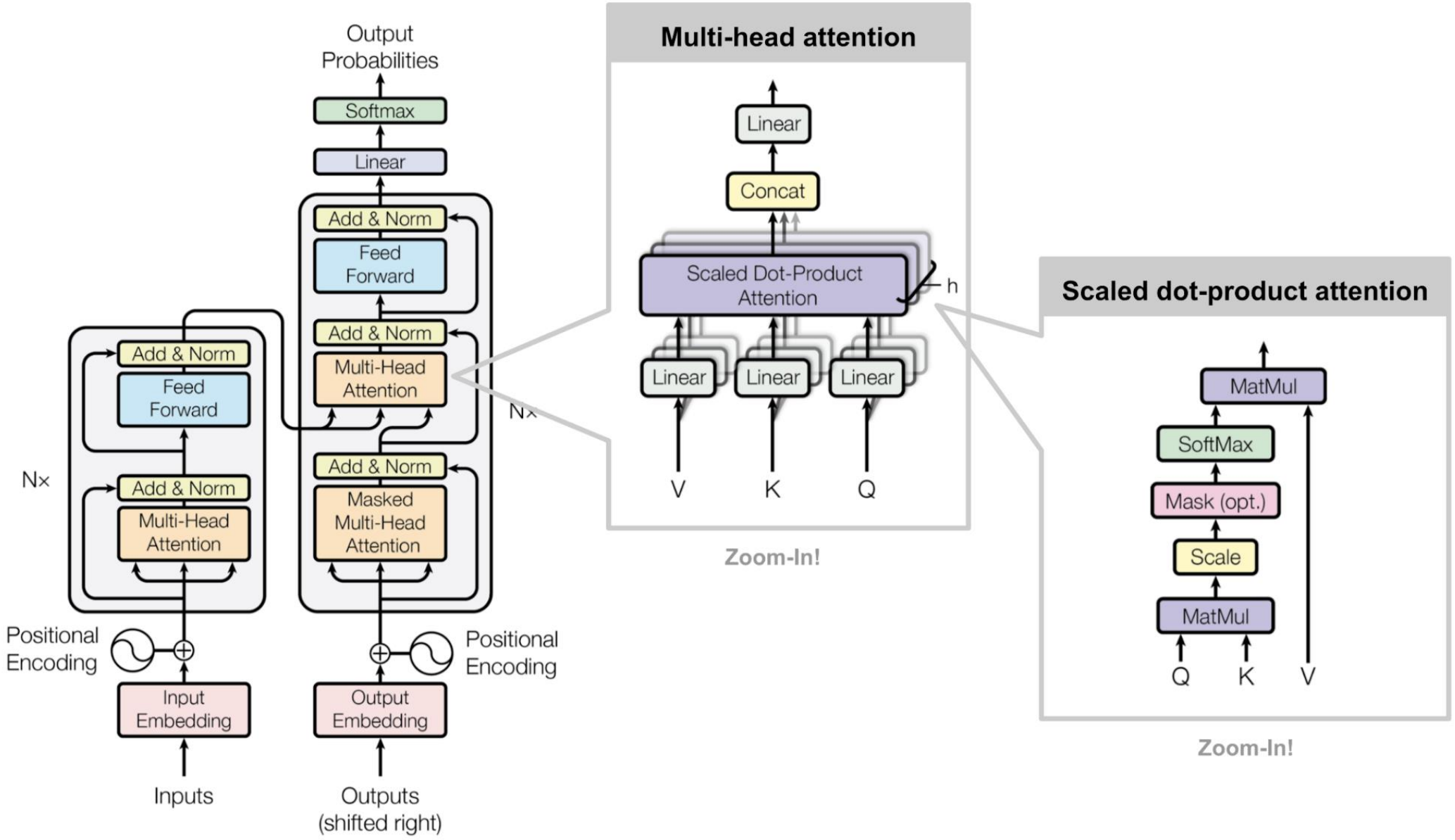
Transformers

Vaswani et al. “Attention Is All You Need”, 2017

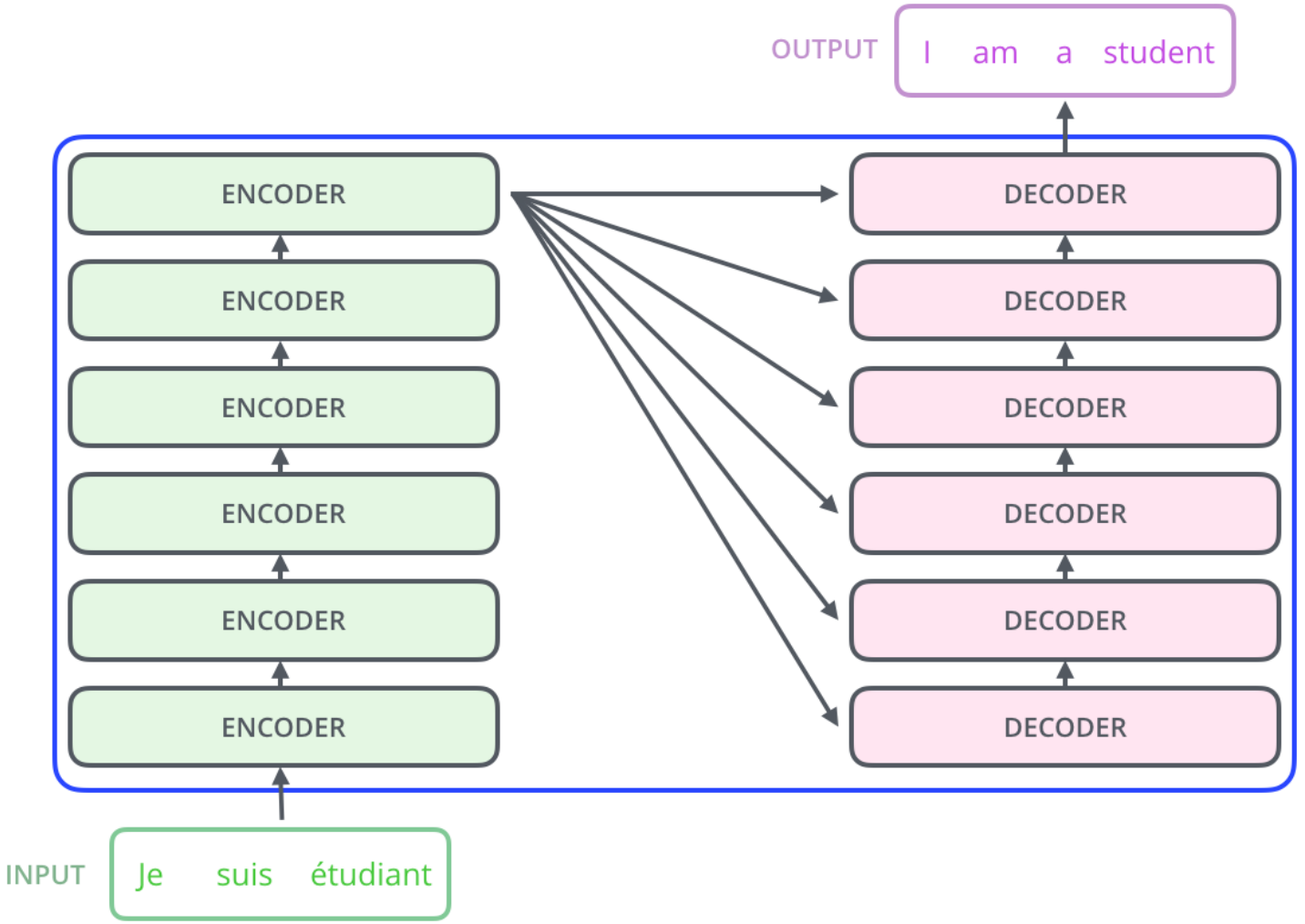


http://jalammar.github.io/illustrated-transformer/

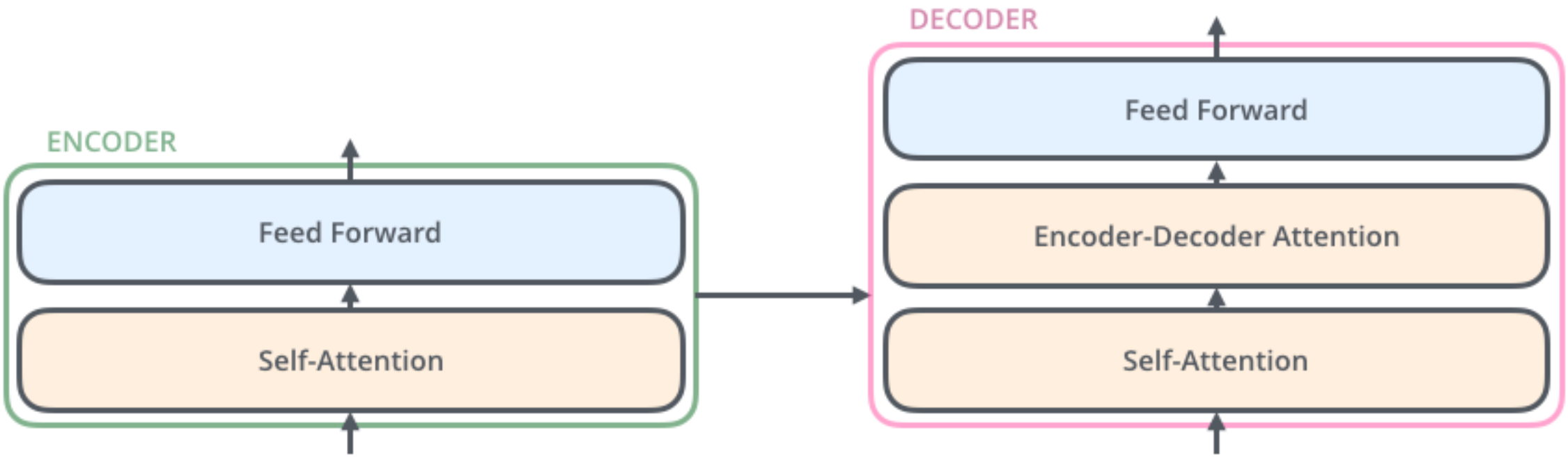
Transformers



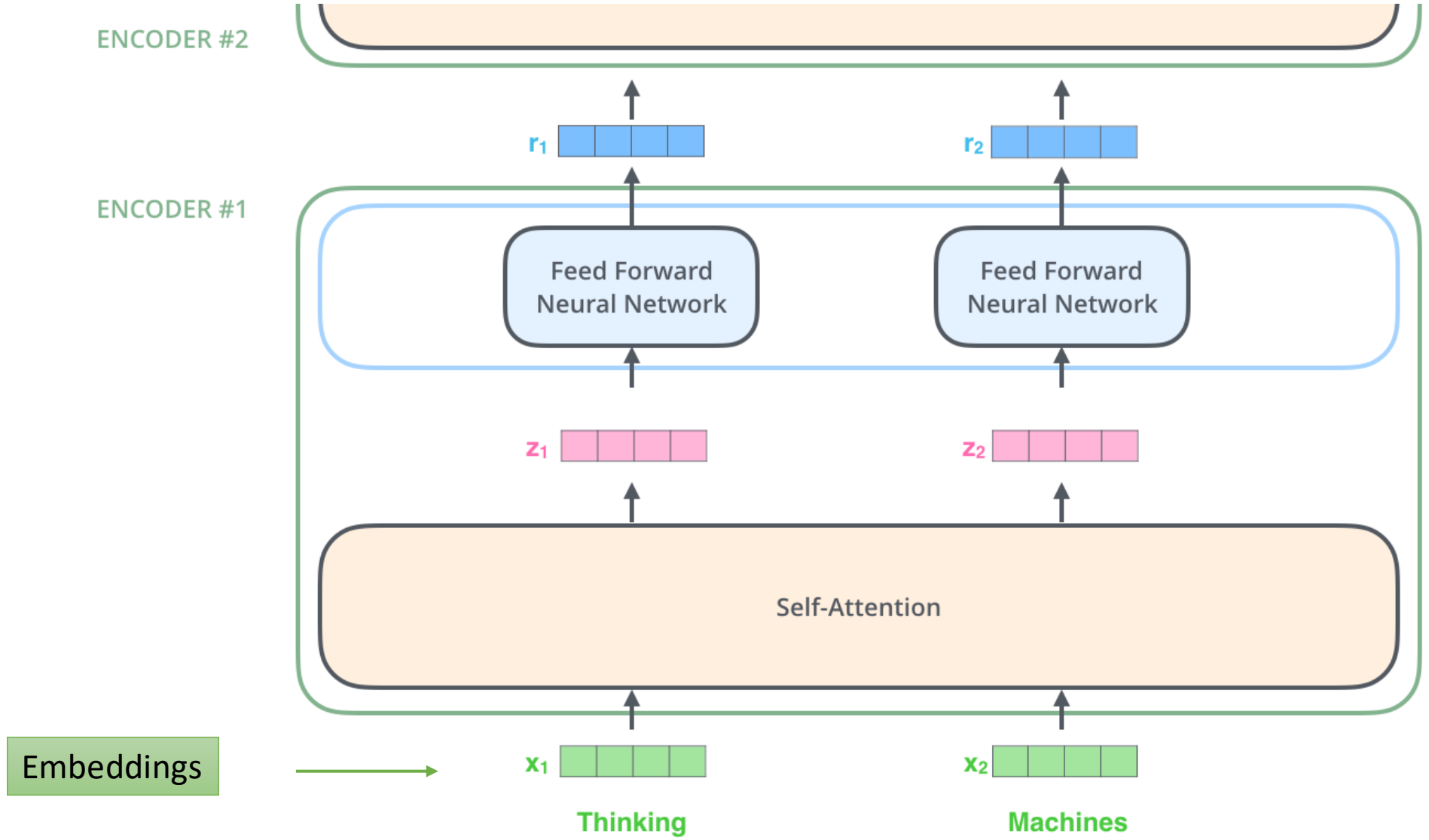
Transformers



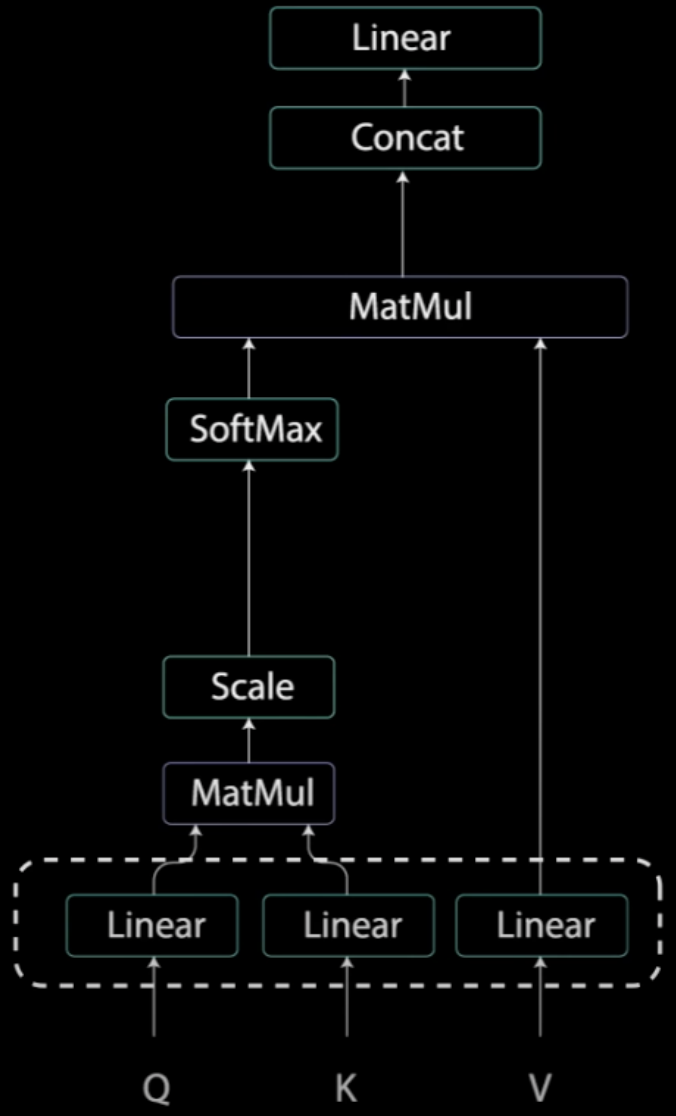
Transformers

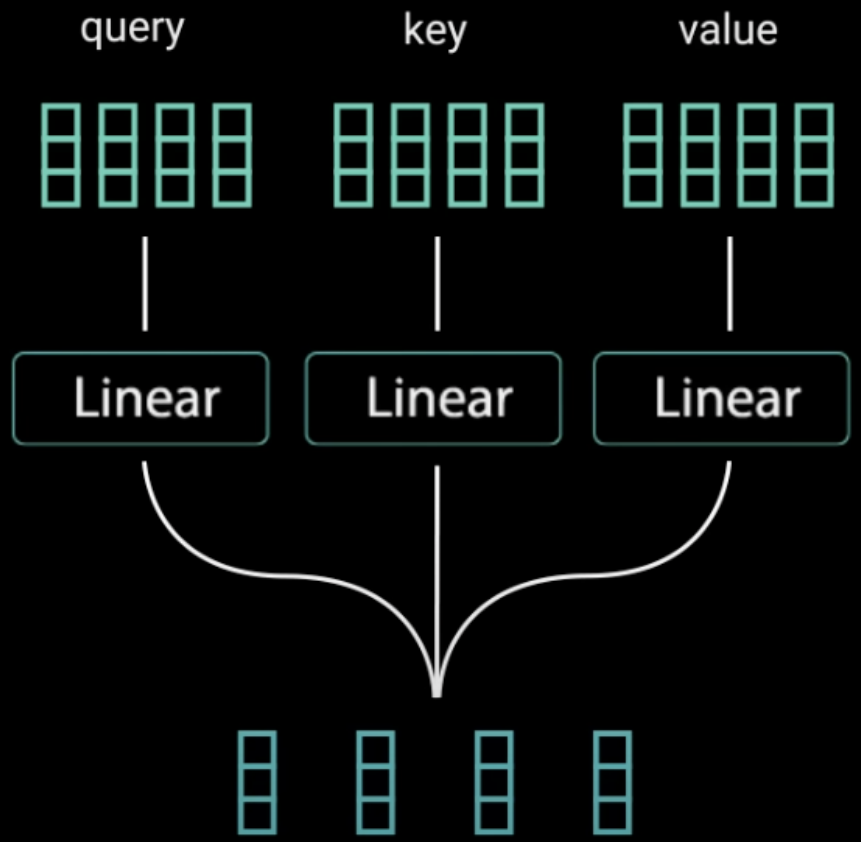


Encoders

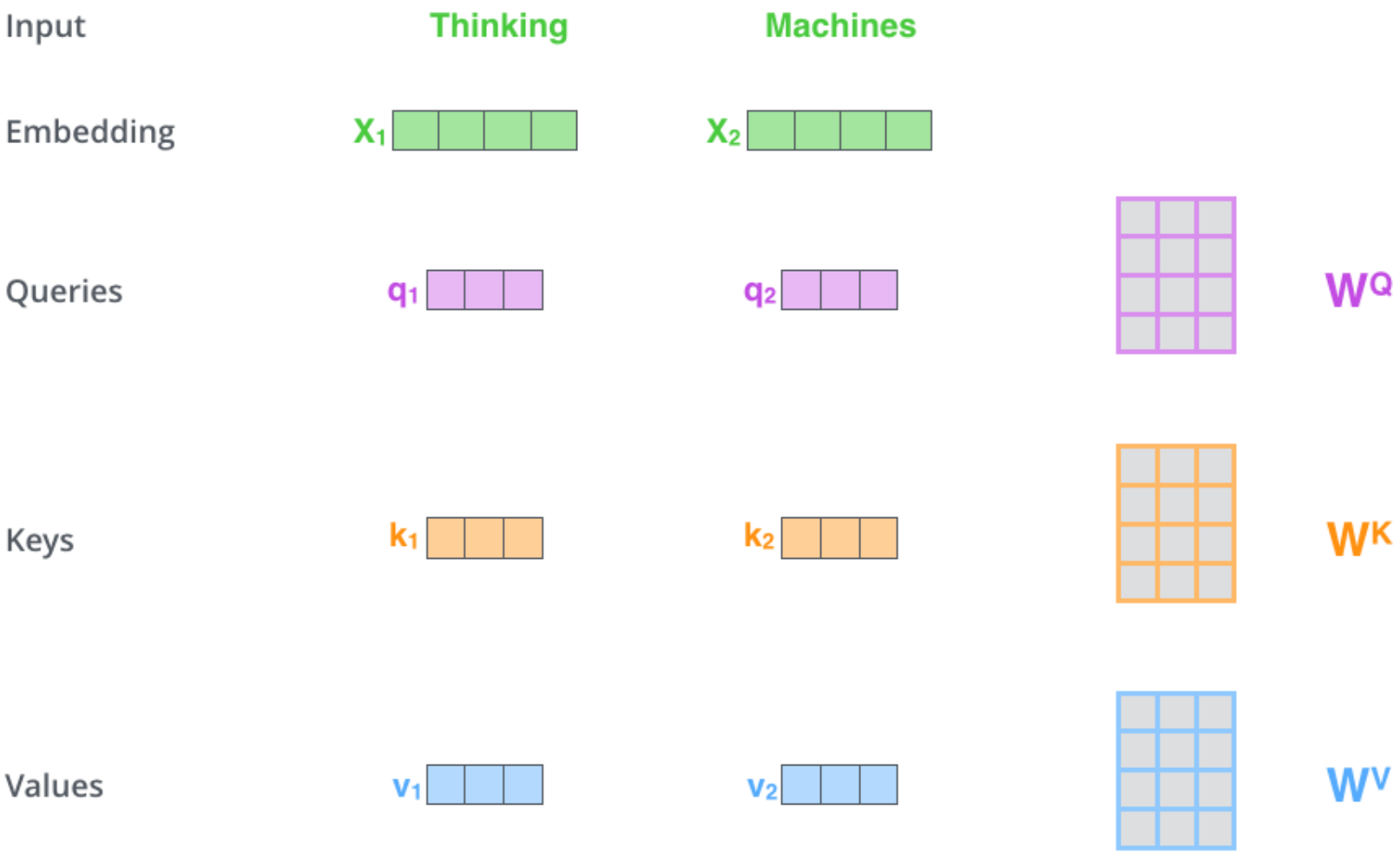

ENCODER #2
ENCODER #1
r1
r2
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Self-Attention
x1
x2
Thinking
Machines
Embeddings

Capa de Auto-Atención

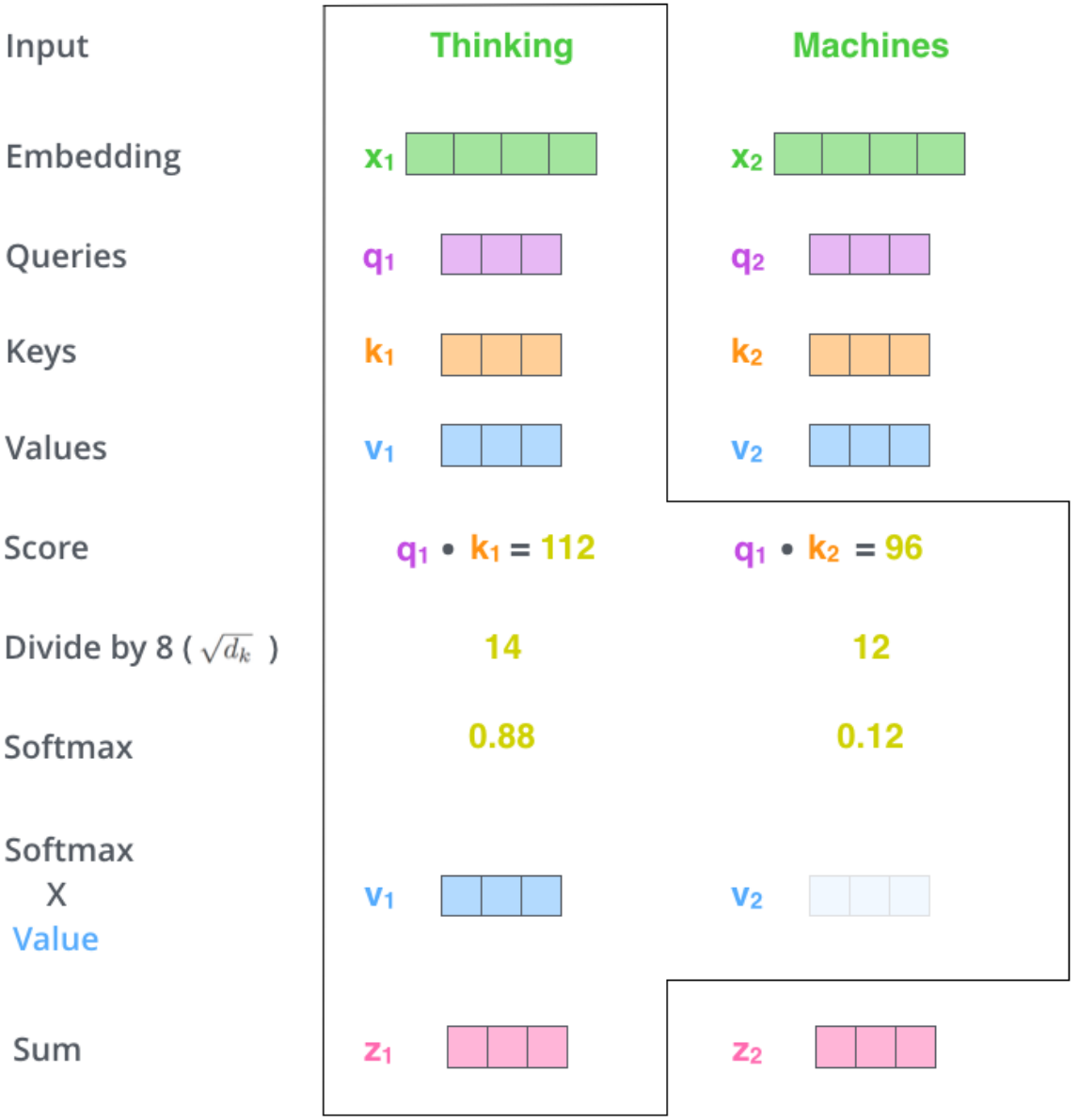




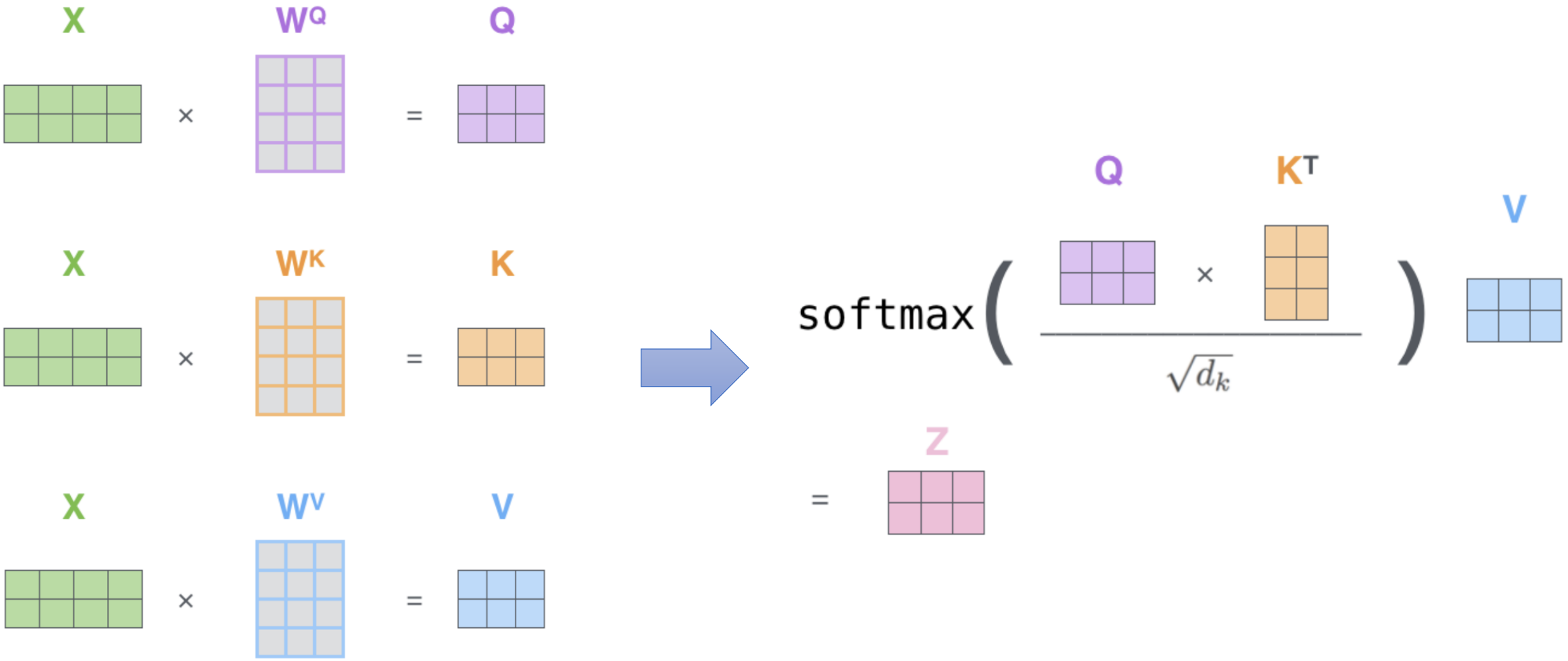
Capa de Auto-Atención



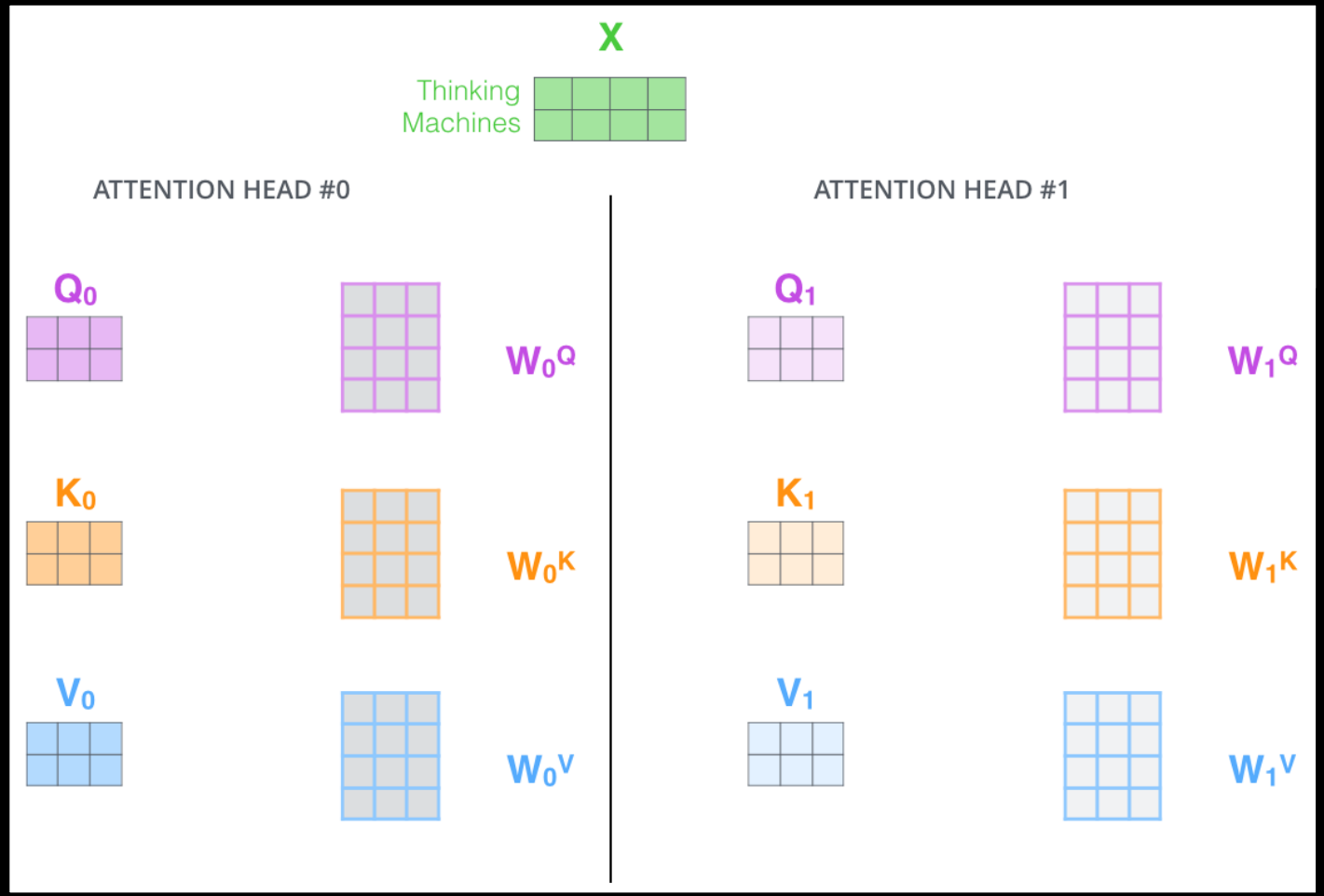
Capa de Auto-Atención

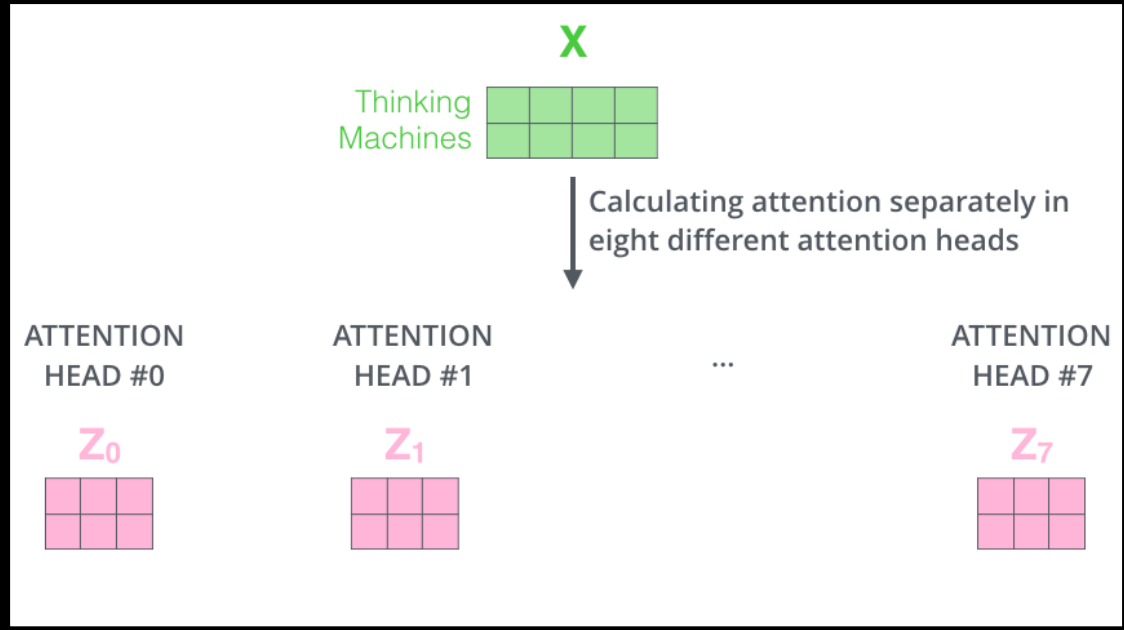


Capa de Auto-Atención (matrices)



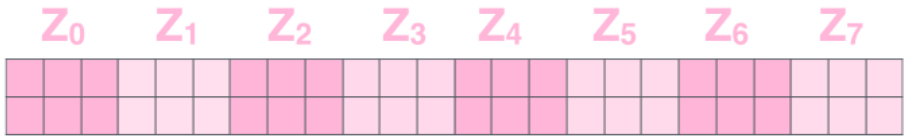
Cabezas de Atención





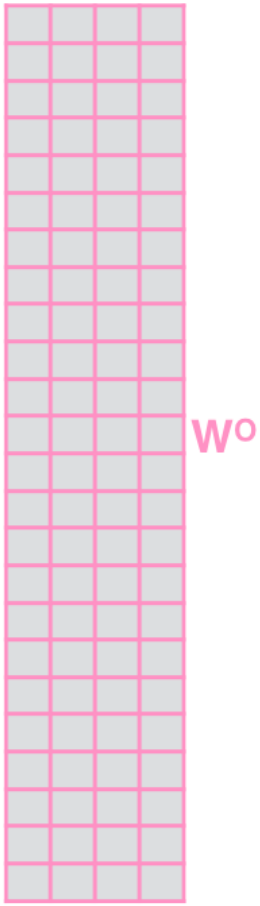
Cabezas de Atención

1) Concatenate all the attention heads



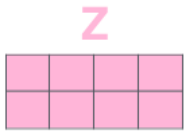
2) Multiply with a weight matrix W^O that was trained jointly with the model

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

=



Todas las matrices en un solo lugar

1) This is our input sentence*

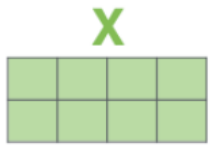
2) We embed each word*

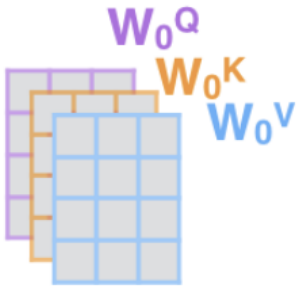
3) Split into 8 heads. We multiply X or R with weight matrices

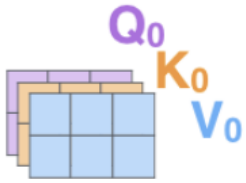
4) Calculate attention using the resulting Q/K/V matrices

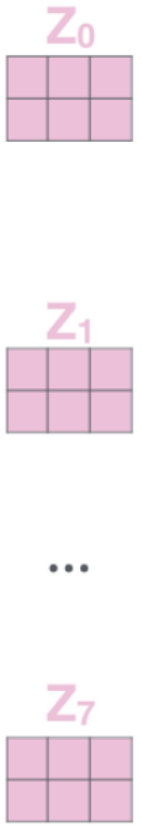
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

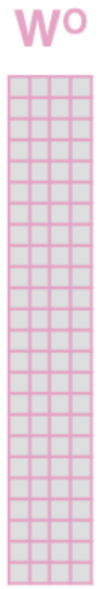
Thinking Machines

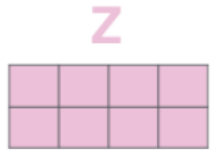






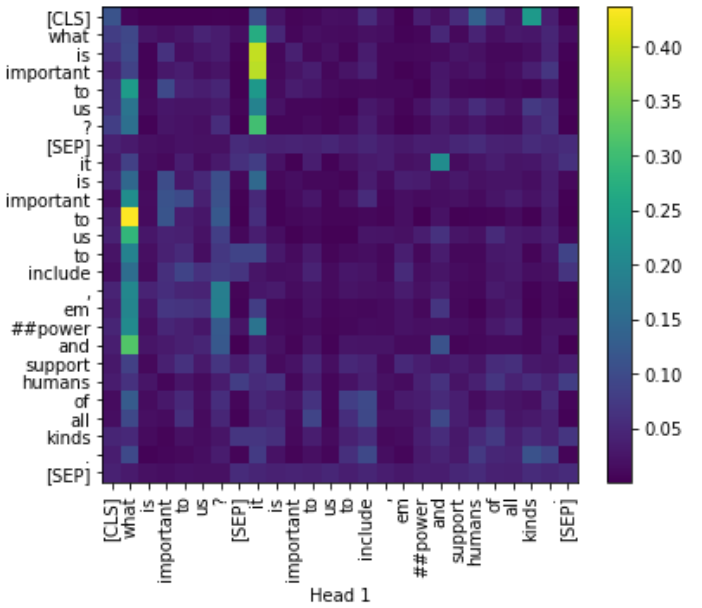


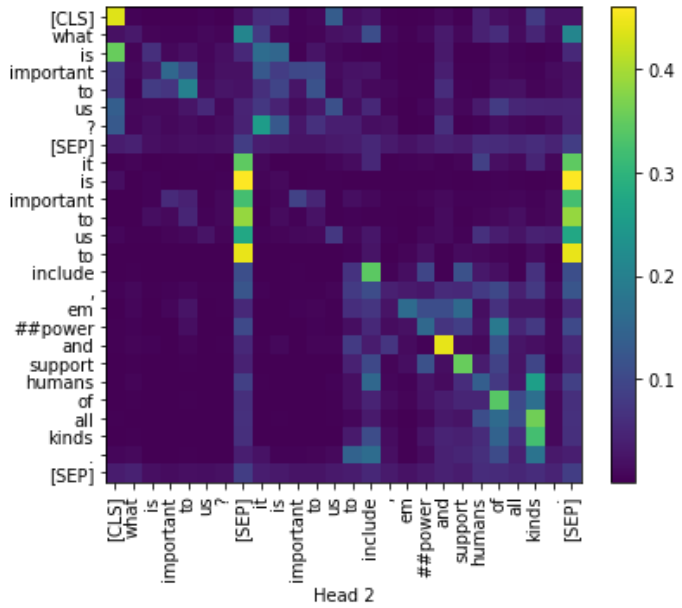


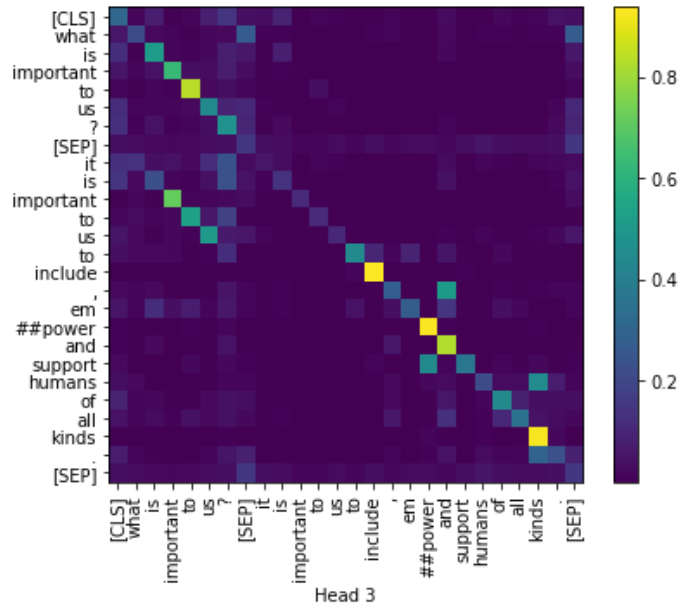


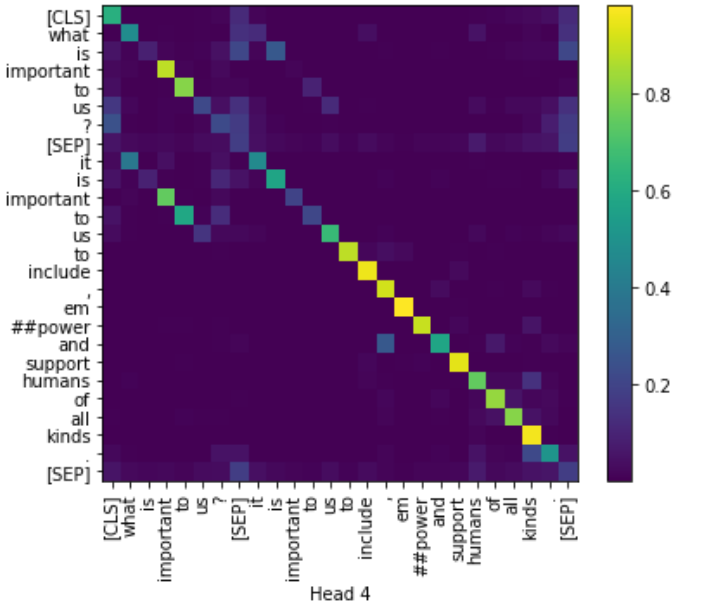
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

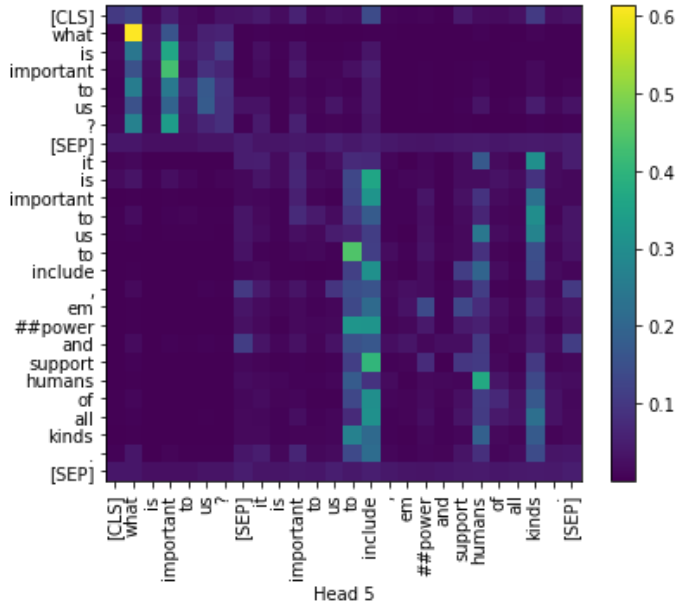
Cabezas de Atención

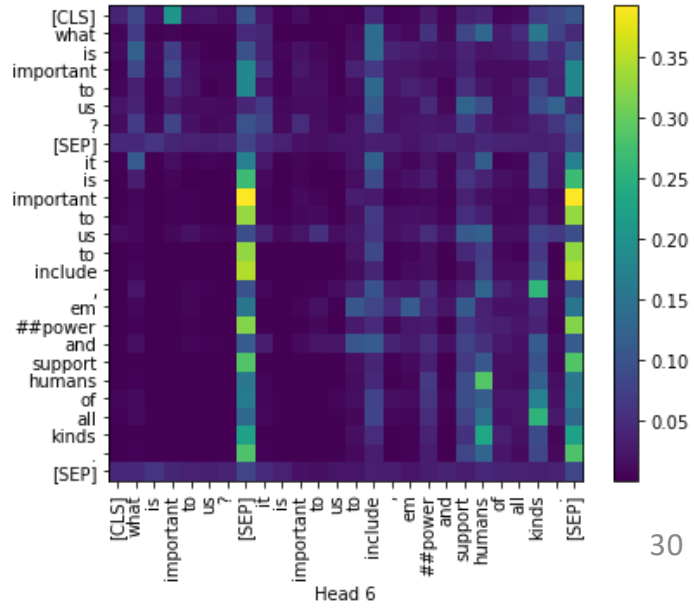




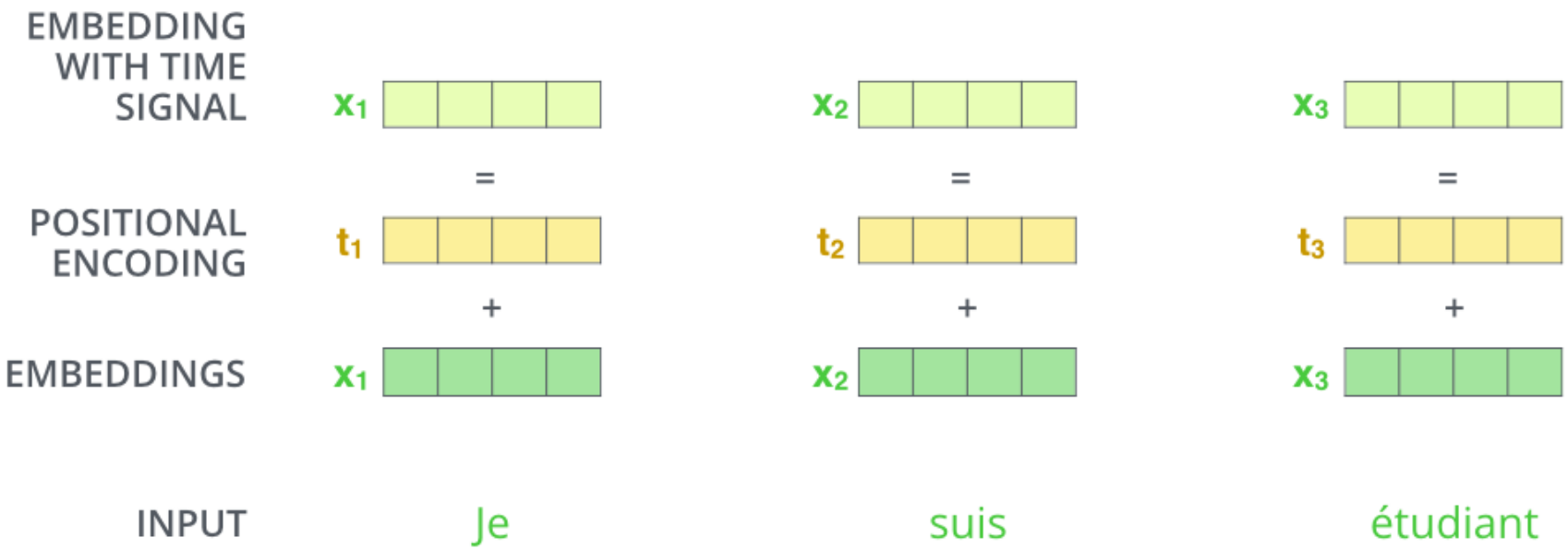




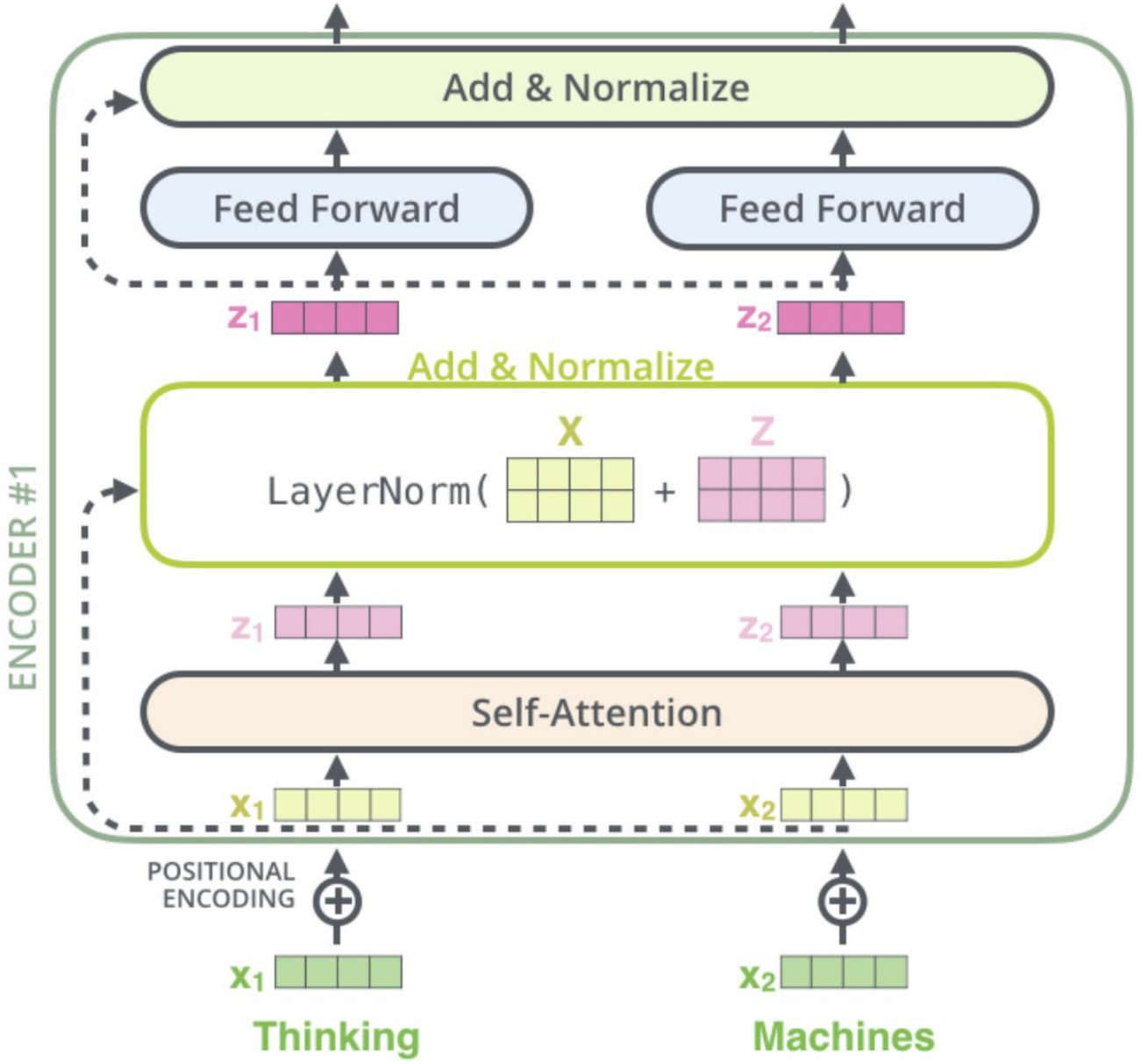




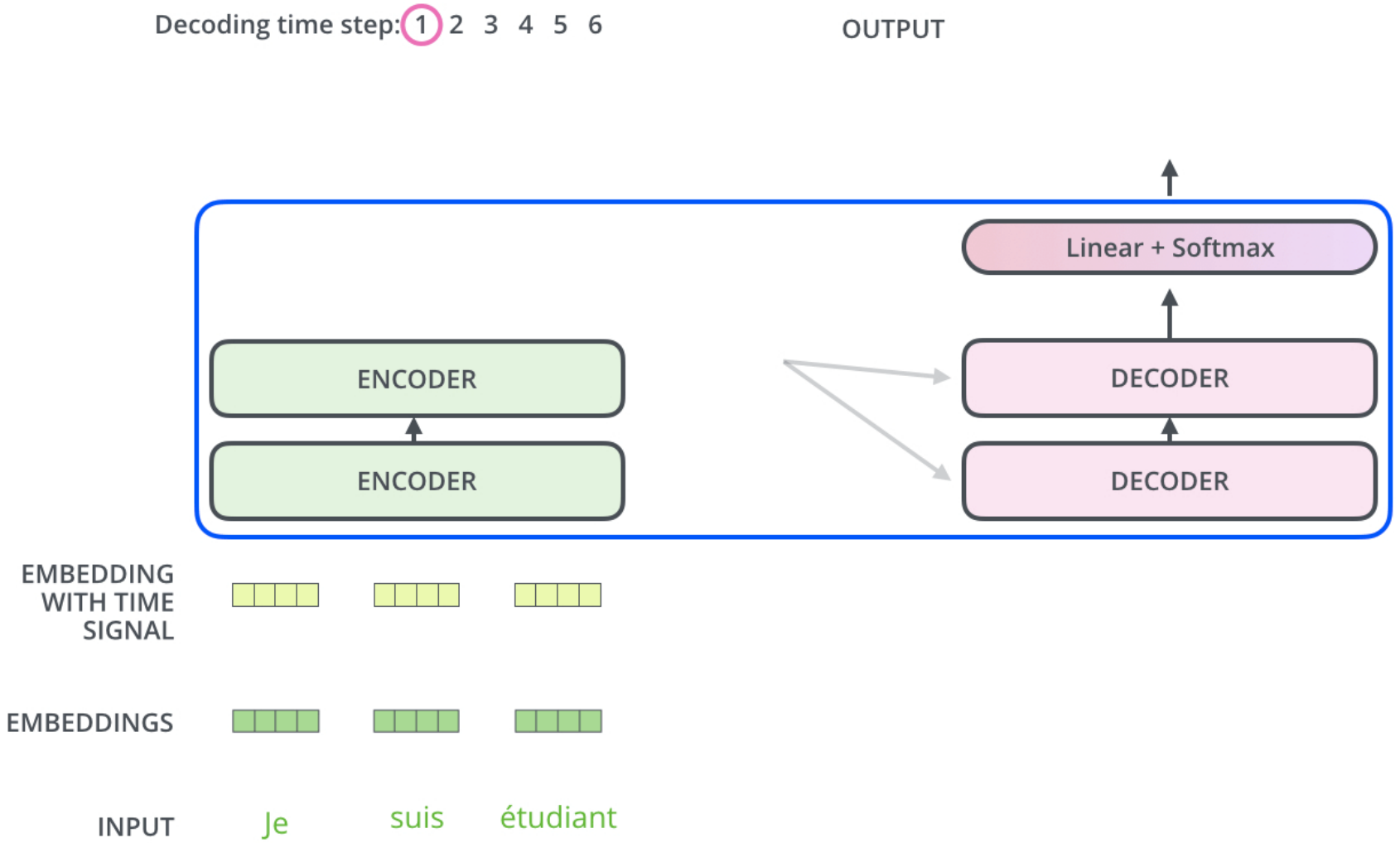
Embeddings Posicionales



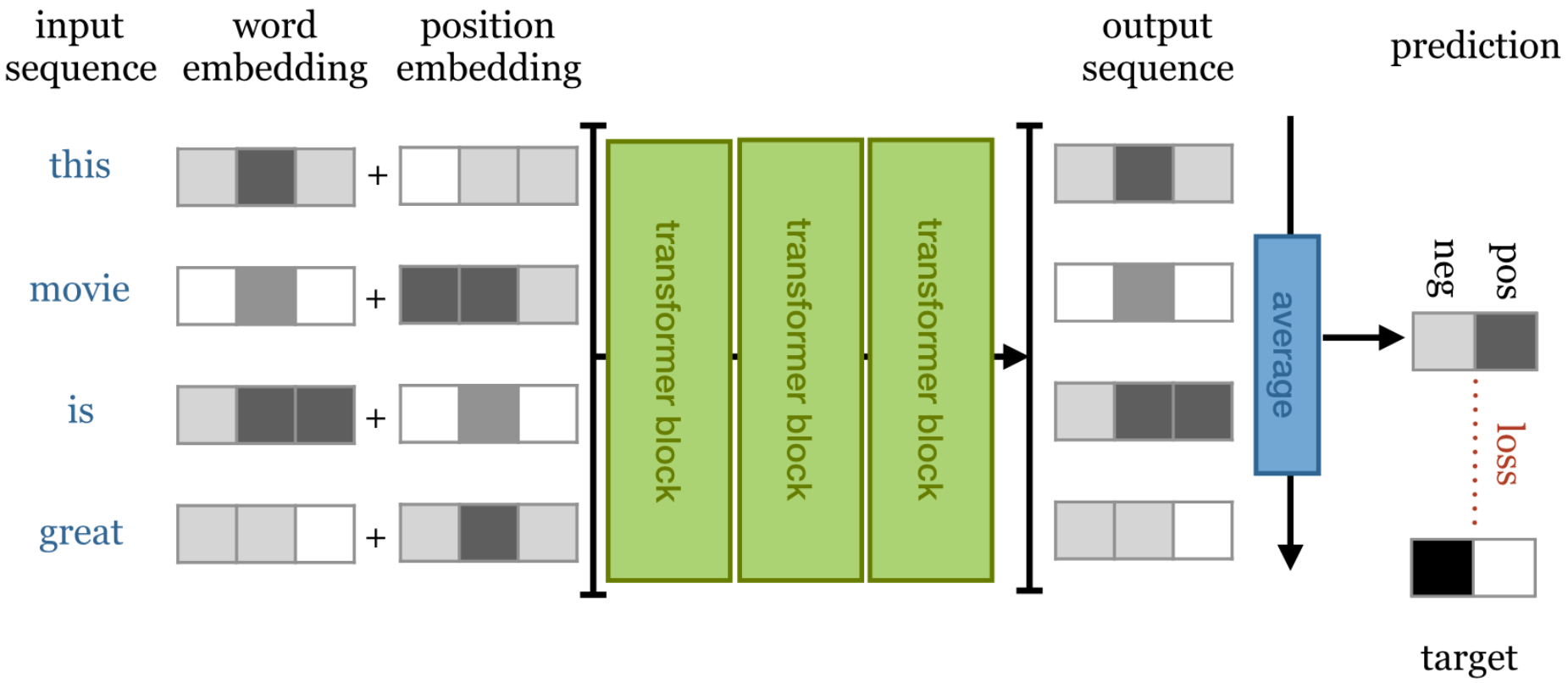
Los residuos



Decoder


Decoding time step: 1 2 3 4 5 6
OUTPUT
Linear + Softmax
ENCODER
ENCODER
DECODER
DECODER
EMBEDDING WITH TIME SIGNAL
EMBEDDINGS
INPUT
Je
suis
étudiant

De Transformers a Clasificadores



https://peterbloem.nl/blog/transformers

De transformadores a clasificadores

- Con los bloques transformadores, la forma más común de construir un clasificador es tener una arquitectura que consta de **una gran cadena de bloques transformadores** .
- Todo lo que necesitamos hacer es descubrir cómo introducir las secuencias de entrada en la arquitectura y cómo transformar la secuencia de salida final en una única **clasificación** .
- El truco del clasificador es aplicar **una combinación de promedio global** a la secuencia de salida final y mapear el resultado a un vector de clase resultado de una función **softmax**.
  - La secuencia de salida se promedia para producir un solo vector (similar a las ***incrustaciones de documentos***).
  - Luego, este vector se proyecta hasta convertirse en un vector con un elemento por clase y se suaviza en probabilidades.

Hugging Face Hub

Hugging Face

- **Hugging Face Hub** es una plataforma abierta y colaborativa que permite compartir, descubrir y reutilizar modelos de aprendizaje automático, conjuntos de datos y espacios de demostración interactivos.
- Albergar miles de **modelos** basados en arquitecturas como BERT, GPT y T5, entre otros, listos para usarse o ajustar en tareas específicas como clasificación de texto, traducción, generación de lenguaje o reconocimiento de entidades.
- La plataforma proporciona **herramientas** como *transformers, datasets y accelerate*, que facilitan el uso de estos recursos en proyectos de investigación o producción.
- Promueve la **transparencia** y la **reproducibilidad** al incluir documentación, licencias, tarjetas de modelo (model cards) y métricas asociadas a cada recurso.

Bibliografía

- Modelos secuencia a secuencia:
  - Cho et al. 2014. Learning Phrase Representations using RNN Encoder–Decoderfor Statistical Machine Translation.
- Atención:
  - Bahdanau et al. 2014. Neural Machine Translation by Jointly Learning to Align and Translate.
  - Vaswani et al. 2017. Attention is All You Need
- Esta clase se basa en:
  - Jay Alammar's blog post: The Illustrated Transformer
  - review of Lilian Weng's Attention? Attention!.