

ATTENTION IS ALL

• YOU NEED.

TRANSFORMER

Evangelina Garza

Victor Gómez

INTRODUCCIÓN

Redes recurrentes.

- Hasta hace poco, las redes recurrentes eran el enfoque estándar para resolver tareas de modelado de secuencias y problemas de traducción.
- En este tipo de modelos se generan estados ocultos h a un tiempo t como función del estado oculto previo a tiempo $t-1$, lo que permite alinear las posiciones de dicha secuencia.
- Algunas desventajas que se presentan:
 - **Vanishing gradient:** Pérdida de información en secuencias largas.
 - **Alto costo computacional.**
 - Su naturaleza secuencial **no permite paralelización.**

INTRODUCCIÓN

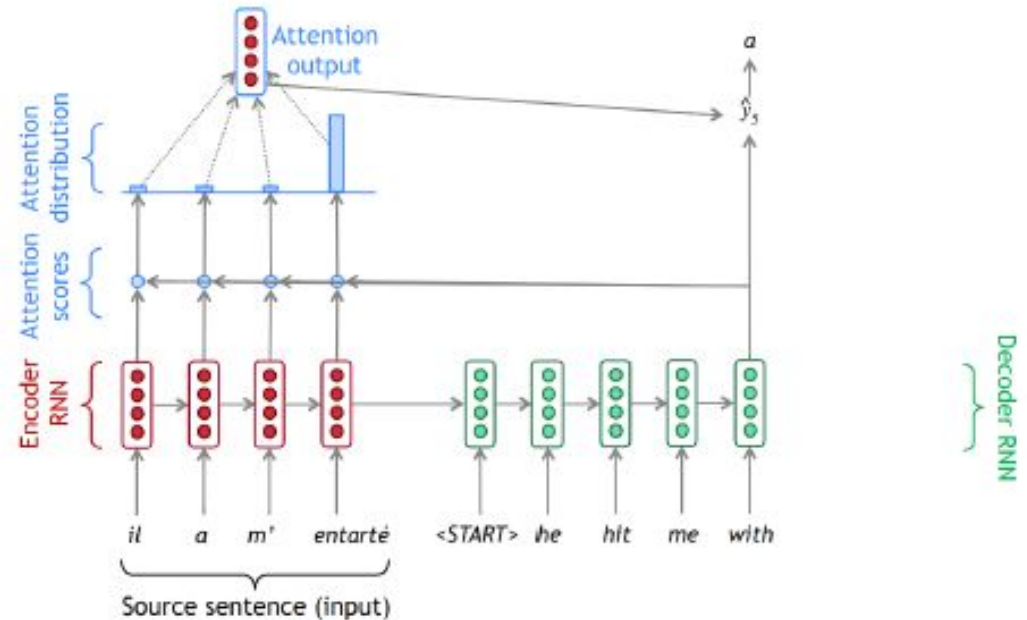
Mecanismos de atención.

- Los mecanismos de atención son eficaces para modelar dependencias sin importar su distancia o posición en la secuencia.
- Esto ha hecho que se conviertan en el estado del arte para atacar problemas de traducción y modelado de secuencias.
- En el 2017, **Vaswani et al.** propusieron una nueva arquitectura de red que utiliza únicamente mecanismos de atención, dejando afuera tanto la recurrencia como la convolución.
- En **"Attention is all you need"** demostraron muy buenos resultados de traducción, altos puntajes BLEU, y menor tiempo de procesamiento vs modelos normalmente utilizados.

INTRODUCCIÓN

Mecanismos de atención.

Los mecanismos de atención se utilizaban normalmente en conjunto con redes recurrentes.

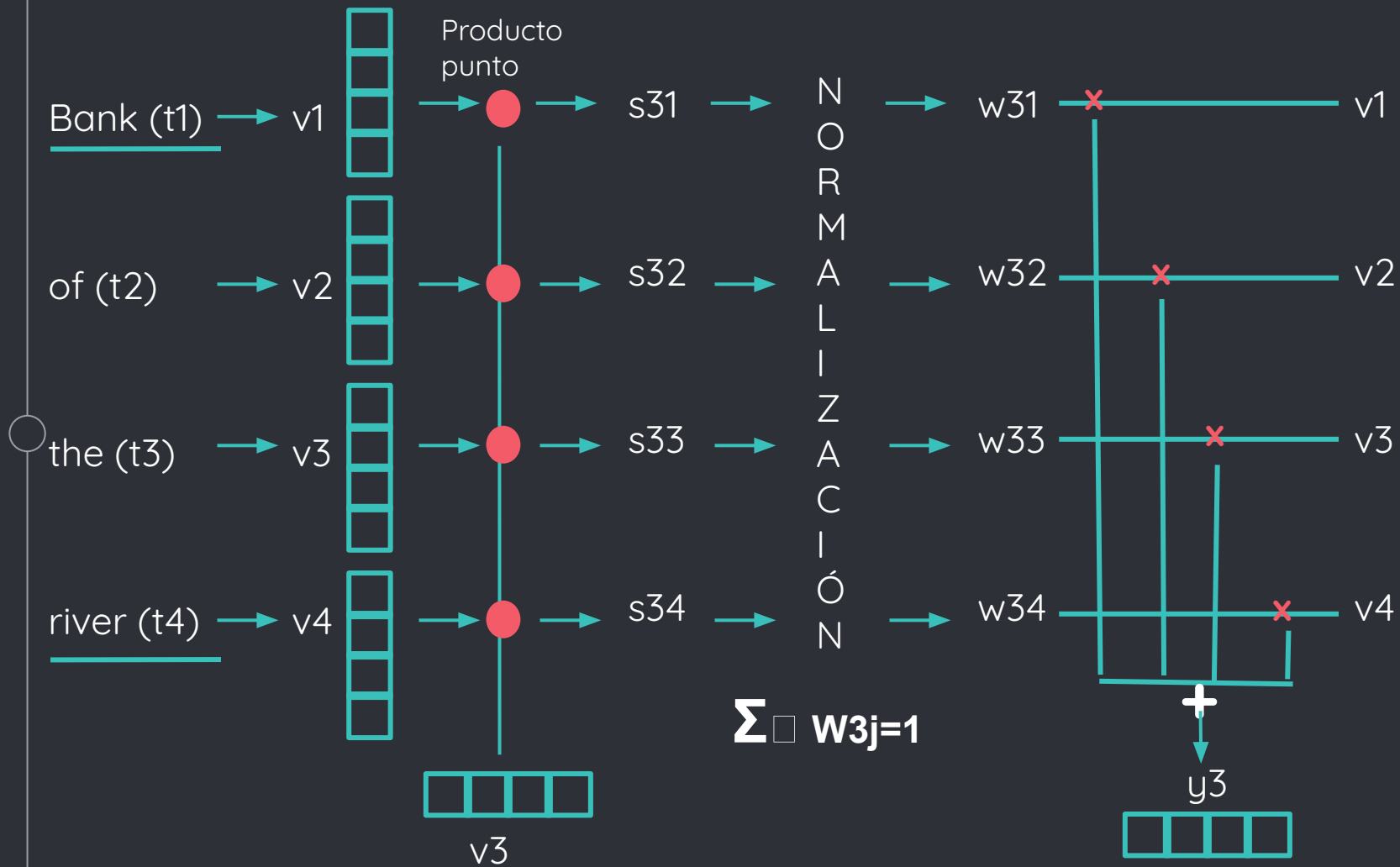


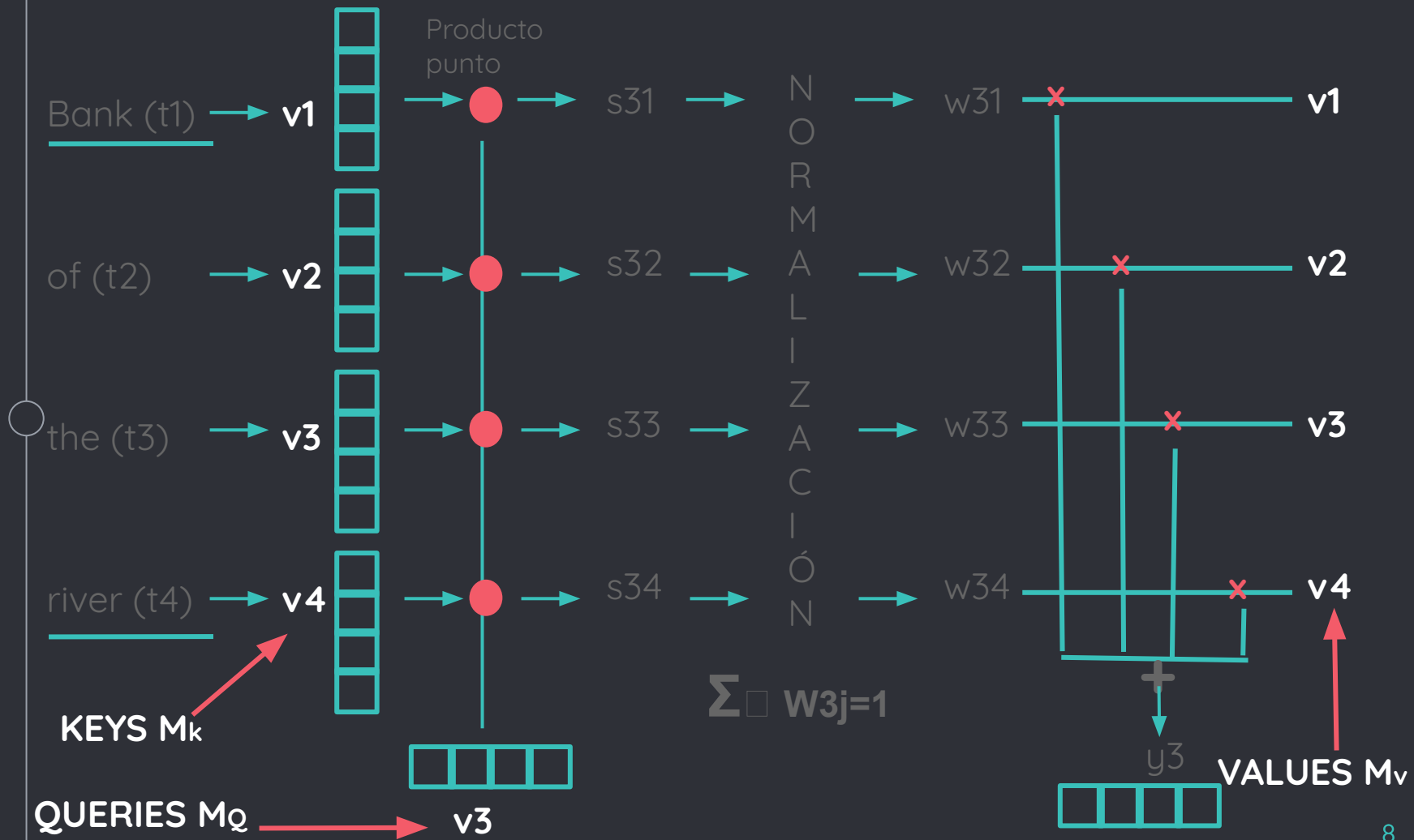
A vertical line on the left side of the slide, with a small circle at its midpoint.

ARQUITECTURA DEL TRANSFORMER

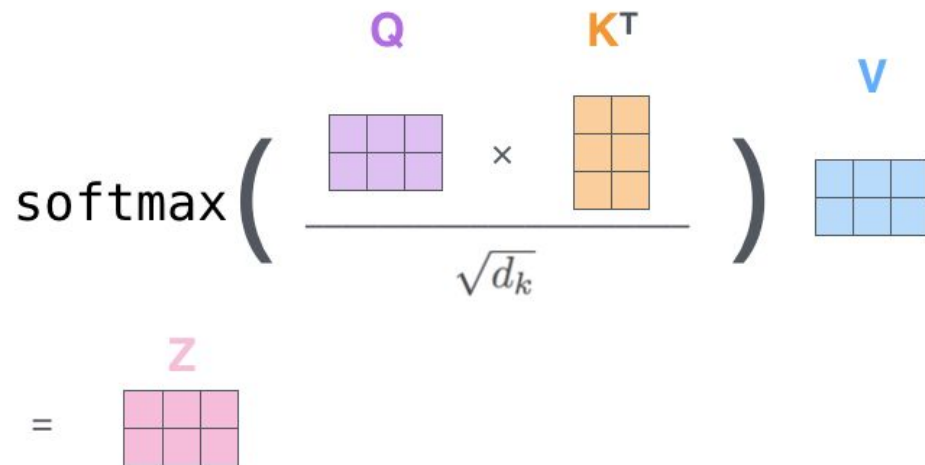
1

MECANISMOS DE AUTO-ATENCIÓN





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



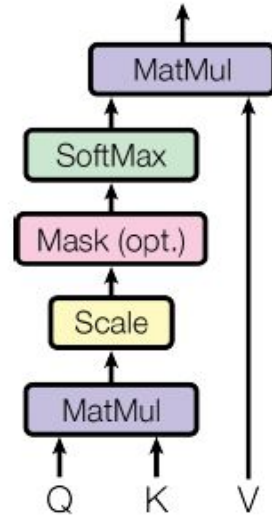
The diagram illustrates the attention mechanism using matrix representations. It shows the calculation of the attention weights using the Query (Q) and Key (K^T) matrices, followed by the multiplication of the resulting matrix with the Value (V) matrix to produce the final output Z.

Q (purple 2x3 grid) × K^T (orange 3x2 grid) = 2x2 grid (divided by $\sqrt{d_k}$)

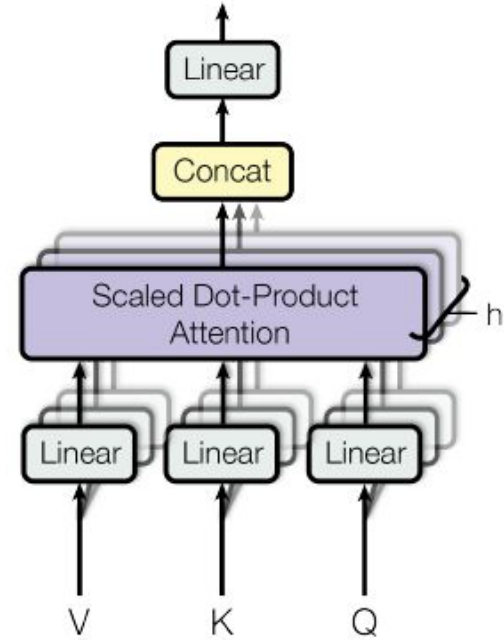
Result (2x2 grid) × V (blue 2x2 grid) = Z (pink 2x2 grid)

Ecuación y representación de las capas de atención utilizadas en la arquitectura del transformer.

Scaled Dot-Product Attention



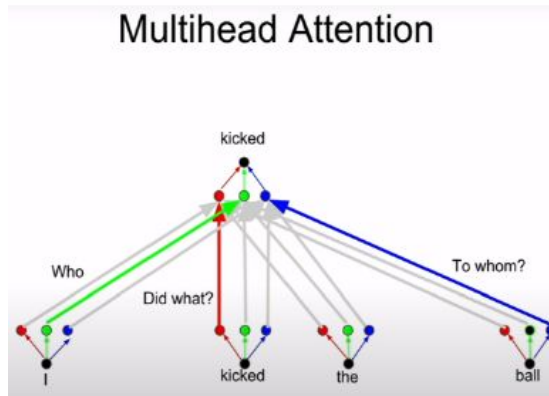
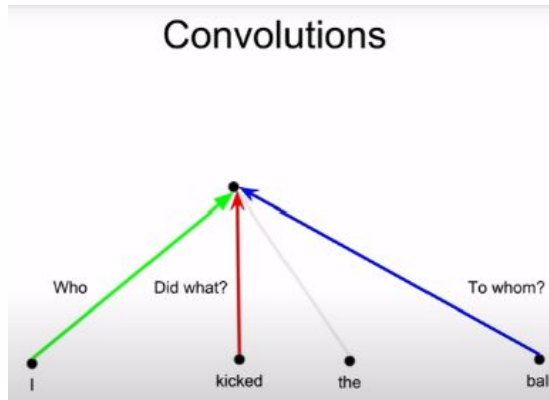
Multi-Head Attention



Esquema de las capas de auto-atención y el multihead utilizado en la arquitectura del transformer. **Vaswani et al. (2017).**

2

AUTO-ATENCIÓN MÚLTIPLE



Representación de la convolución y del mecanismo de atención múltiple. **(Stanford, 2019)**

$$MultiHead(Q, K, V) = Concat(head_i, \dots, head_h) W^O$$

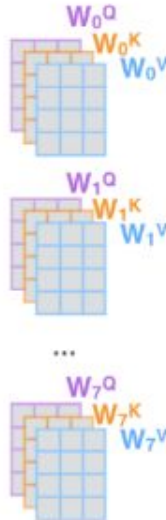
1) This is our
input sentence*

Thinking
Machines

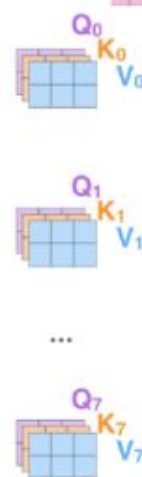
2) We embed
each word*



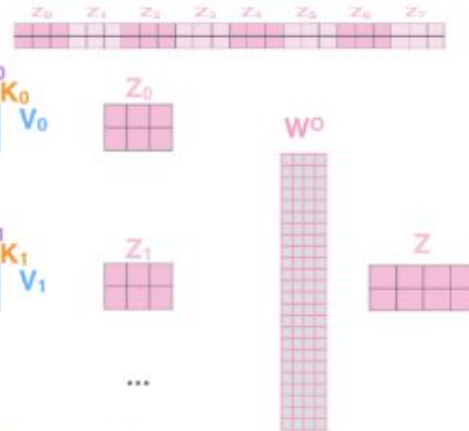
3) Split into 8 heads.
We multiply X or
 R with weight matrices



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



Representación del mecanismo de atención múltiple. (Alammar, 2018)

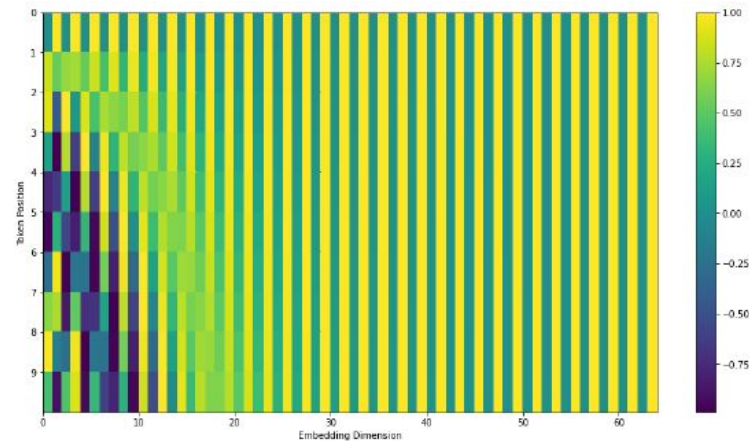
3

ENTRADAS Y SALIDAS DEL TRANSFORMER

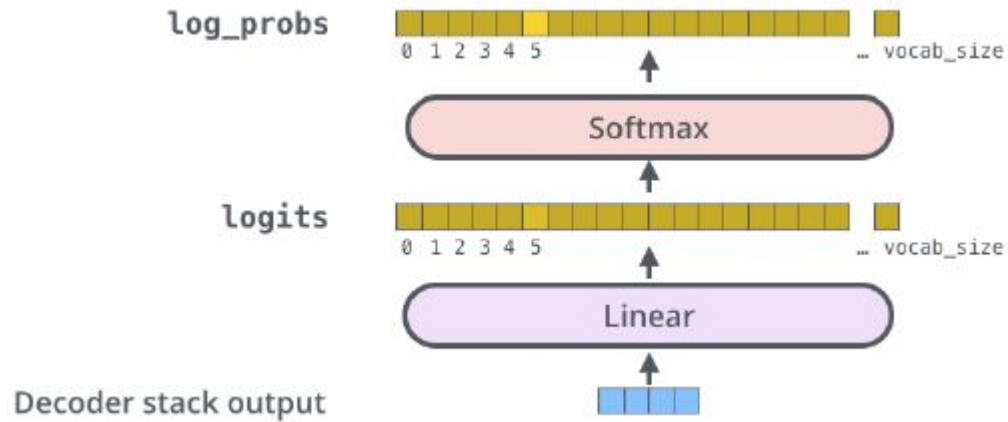


$$PE_{(pos, 2i)} = \sin\left(pos / 10000^{2i/d_m}\right)$$

$$PPE_{(pos, 2i+1)} = \cos\left(pos / 10000^{2i/d_m}\right)$$



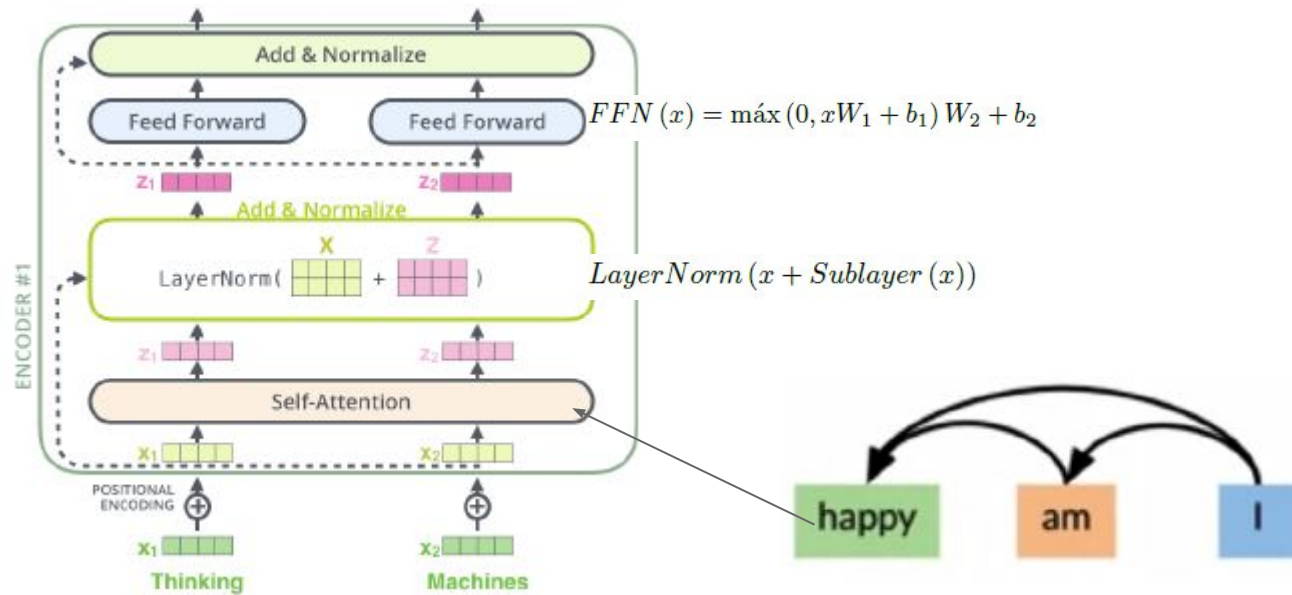
Entradas al transformer, patron de posición. (Alammar, 2018)



Salida del transformer. (Alammar, 2018)

4

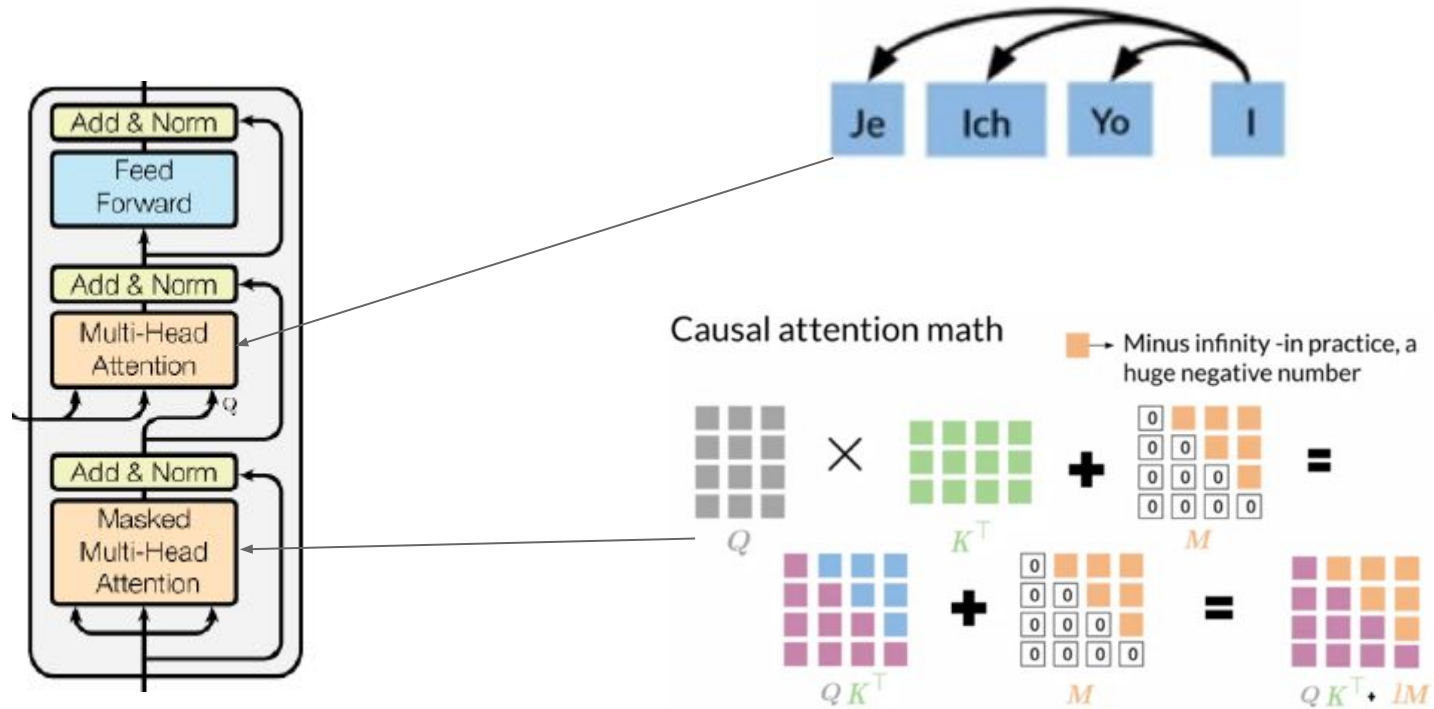
CODIFICADOR DEL TRANSFORMER



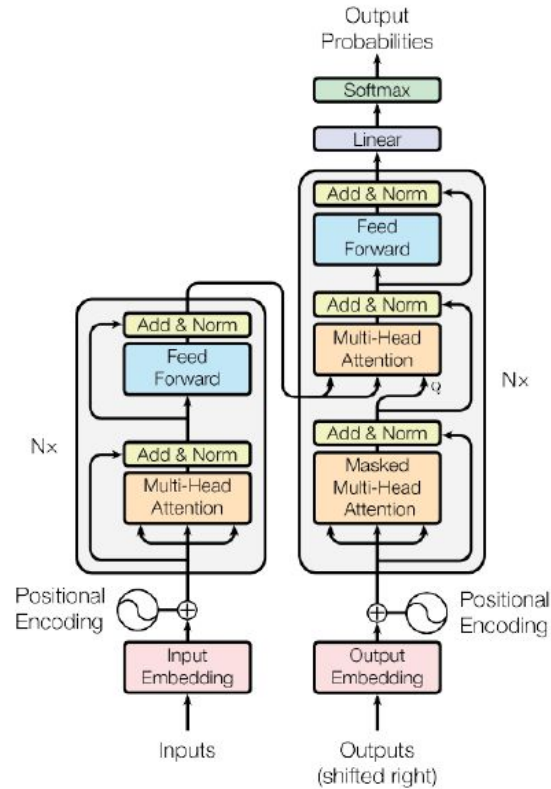
Subcapas del codificador. (Alammar 2018). Auto atención en el codificador (Mourri et al.)

5

DECODIFICADOR DEL TRANSFORMER



Subcapas del decodificador (**Vaswani et al., 2017**). Auto atención en la primer sub capa del decodificador con mascara, Auto atención en la segunda subcapa del codificador (**Mourri et al.**)



Transformer. (Vaswani et al., 2017).

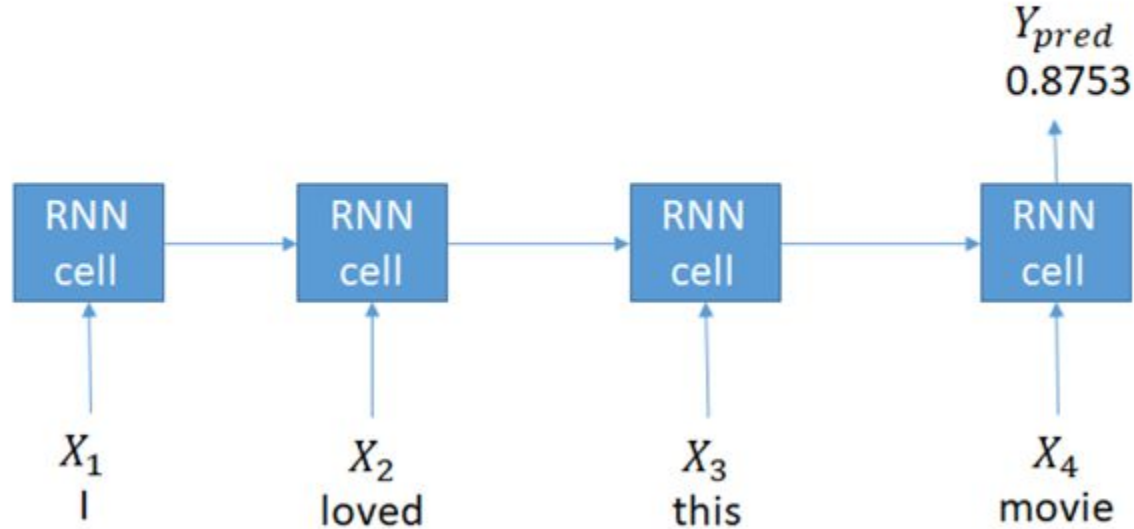
A vertical line on the left side of the slide, with a small circle at the level of the title.

APLICACIONES A NLP E IMÁGENES

6

ANÁLISIS DE SENTIMIENTOS EN TEXTO

Many to one: classification



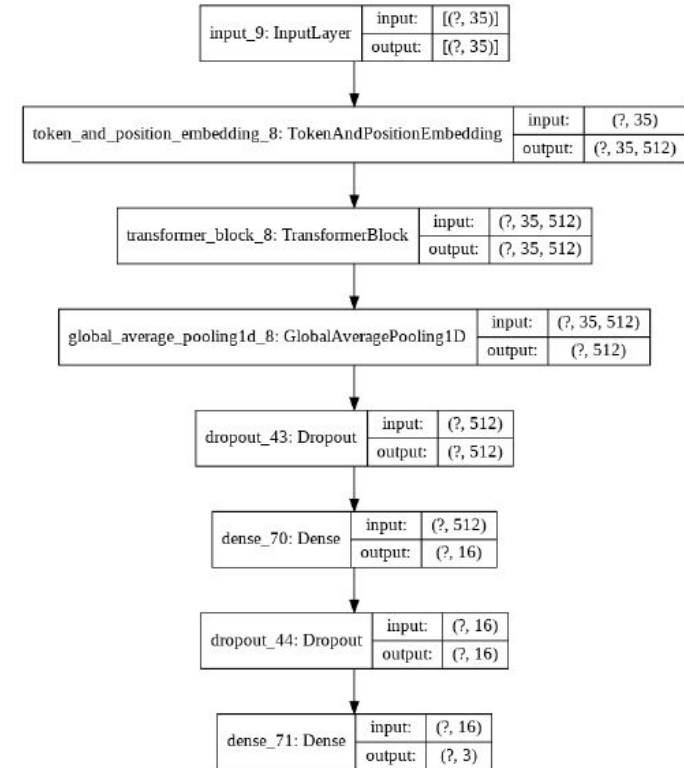
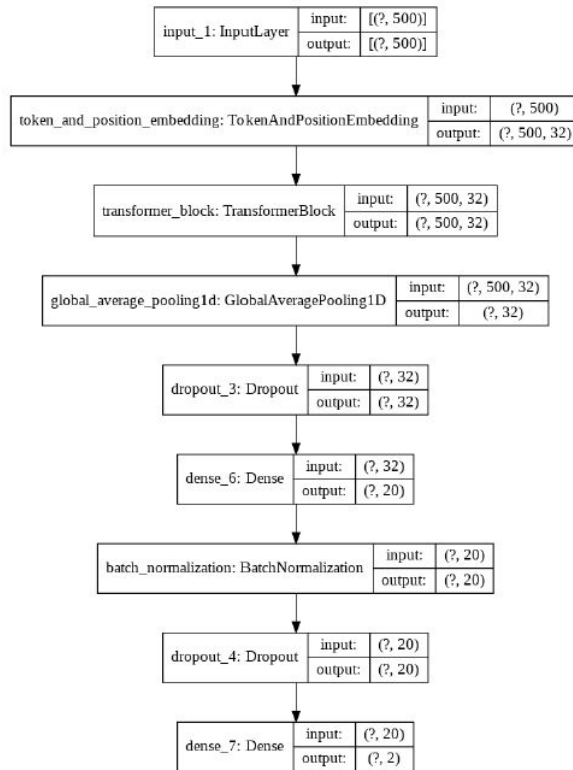
Análisis de sentimientos, modelo secuencia a secuencia tipo muchos a uno. **(Cecchini, datacamp).**

Textos en inglés

- Conjunto de datos: IMDB, Críticas de películas.
- Clasificación binaria (0,1)
- Clases balanceadas
- Entrenamiento: 25,000 datos
- Prueba: 25,000 datos
- Vocabulario: 10,000 palabras
- Tamaños secuencia: 500
- Padding: 0

Textos en español de México

- Conjunto de datos: TASS, Tweets.
- Clasificación multi clase (P,N,NEU)
- Clases balanceadas
- Entrenamiento: 3,103 datos
- Prueba: 776 datos
- Vocabulario: 40,000 palabras
- Tamaños secuencia: 35
- Padding: 0
- Pre proceso al texto: Minusculas, acentos, duplicados, #, 123..., @, links



Arquitecturas de modelos basados en el codificador del transformer. a) para textos en inglés, b) para textos en español.

	<i>Train. Param</i>	<i>Epochs</i>	<i>Avg t/epoch</i>	<i>Val-Accuracy</i>
<i>1X RNN bid</i>	338,291	5	35s	0.8906
<i>2X CNN 1D</i>	1,315,937	10	15s	0.8718
<i>1X Encoder</i>	327,206	1	32s	0.8872

Resultados para los textos en inglés.

	<i>Train. Param</i>	<i>Epochs</i>	<i>Avg t/epoch</i>	<i>Val-Accuracy</i>
<i>Vader</i>	-	-	-	0.4
<i>HuggingFace</i>	-	-	-	0.36
<i>SVM-TFIDF</i>	-	-	-	0.64
<i>1X RNN bid</i>	20,620,611	8	22s	0.63
<i>1X CNN 1D</i>	20,628,067	24	20s	0.64
<i>1X Encoder</i>	21,557,843	12	11s	0.62

Resultados para los textos en español.

7

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE
RECOGNITION AT SCALE

● VISION TRANSFORMER

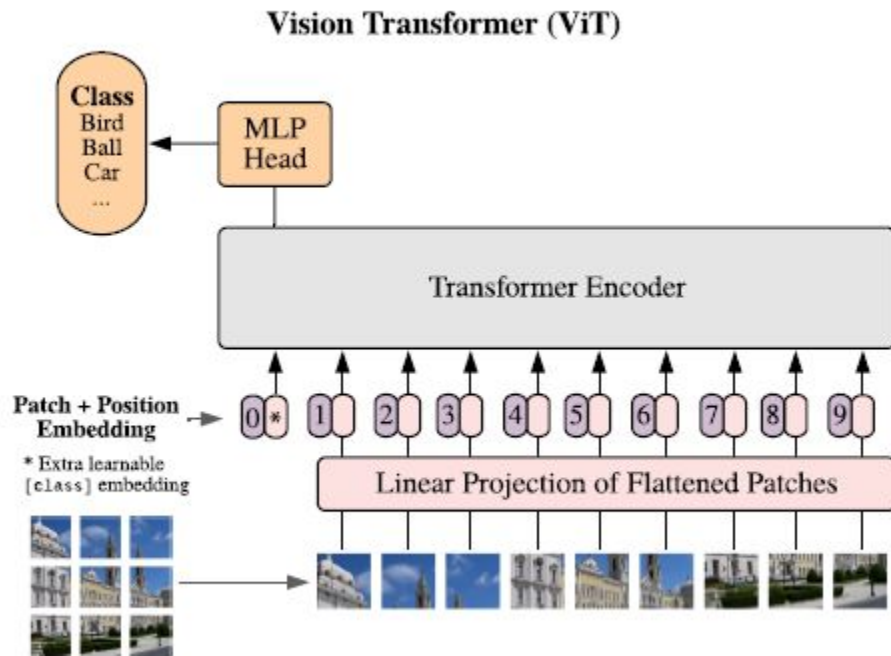
○ Se toma una imagen (de dimensiones $H \times W \times C$) y se transforma en una secuencia de parches en dos dimensiones (de dimensiones $P \times P$).

En donde $H \times W$ es la resolución de la imagen original, C son los canales, $P \times P$ es la resolución de los parches y $N = HW/P^2$ el número de parches resultantes.

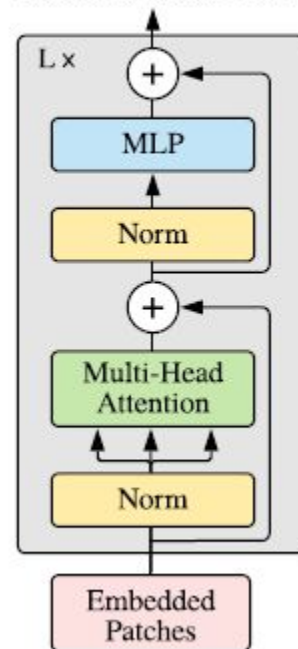
Se "aplanan" (**flatten**) luego estos parches y se mapean a D dimensiones con una proyección lineal entrenada, obteniendo así los embeddings de los parches.

Se agregan los **embeddings posicionales**, que están también en una dimensión y el embedding resultante es el que se utiliza de entrada al codificador del transformer.

VISION TRANSFORMER



Transformer Encoder



VIT vs RESNET

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 \pm 0.04	87.76 \pm 0.03	85.30 \pm 0.02	87.54 \pm 0.02	88.4/88.5*
ImageNet ReaL	90.72 \pm 0.05	90.54 \pm 0.03	88.62 \pm 0.05	90.54	90.55
CIFAR-10	99.50 \pm 0.06	99.42 \pm 0.03	99.15 \pm 0.03	99.37 \pm 0.06	—
CIFAR-100	94.55 \pm 0.04	93.90 \pm 0.05	93.25 \pm 0.05	93.51 \pm 0.08	—
Oxford-IIIT Pets	97.56 \pm 0.03	97.32 \pm 0.11	94.67 \pm 0.15	96.62 \pm 0.23	—
Oxford Flowers-102	99.68 \pm 0.02	99.74 \pm 0.00	99.61 \pm 0.02	99.63 \pm 0.03	—
VTAB (19 tasks)	77.63 \pm 0.23	76.28 \pm 0.46	72.72 \pm 0.21	76.29 \pm 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

2020-12-12 04:04:14 (156 KB/s) - 'picsum.jpg' saved [22173]

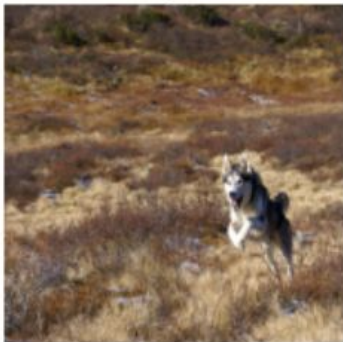


```
0.19203 : alp
0.13293 : cliff, drop, drop-off
0.12670 : valley, vale
0.07687 : seashore, coast, seacoast, sea-coast
0.07473 : volcano
0.05703 : balloon
0.03229 : bell_cote, bell_cot
0.02437 : lakeside, lakeshore
0.02322 : beacon, lighthouse, beacon_light, pharos
0.02319 : promontory, headland, head, foreland
```

Ejemplo de clasificación de imagen utilizando el Vision Transformer. Implementación original de **Dosovitskiy et al.** en **JAX**.

MAPAS DE ATENCIÓN

Original



Attention Map



Prediction: Eskimo dog, husky

Original



Attention Map



Prediction: promontory, headland, head, foreland

Original



Attention Map



Prediction: Pembroke, Pembroke Welsh corgi

Original



Attention Map



Prediction: Chihuahua

8

CONCLUSIONES y REFERENCIAS

● CONCLUSIONES

- Se encontró que los modelos de Transformers son muy **sensibles al tamaño de la secuencia**, para secuencias muy largas tienden a fallar.
- Además de que aún son **poco maduros** y **no se encuentran disponibles en todas las librerías** y algunas de estas implementaciones pueden arrojar **muchos errores de ejecución o compatibilidad**.
- A pesar de lo anterior, para las tareas **en NLP en análisis de sentimientos**, **los resultados compiten** claramente con los obtenidos **con los modelos de redes recurrentes y convolucionales**, con la **ventaja** de que su **entrenamiento** es considerablemente **más rápido**.

CONCLUSIONES

- En el caso del **Vision Transformer**, el modelo obtenía resultados bastante **malos** cuando se entrenaba con conjuntos de datos de tamaño **chico o mediano**. Cuando se entrenó en **base de datos grandes** (más de 13 millones de imágenes) se llegó a una precisión comparable o **mejor que los modelos en el estado del arte (ResNet)**.
- También se presentaron muchos **errores de compatibilidad**, específicamente de las librerías en las que se ha implementado el Vision Transformer hasta ahora.
- El **poder computacional** requerido para entrenar los modelos en grandes bases de datos es sólo una pequeña fracción del poder computacional con que se cuenta actualmente.
- Además de que el procesamiento es **paralelizable** por lo que mayor poder computacional no significa, en este caso, mayor tiempo de entrenamiento.

REFERENCIAS

- Alammar, J. (2018), "The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.," *Github*, Available at <http://jalammar.github.io/illustrated-transformer/>.
- Chollet, F. (2017), *Deep Learning with Python, 2018 21st International Conference on Information Fusion, FUSION 2018*, Manning Publications Co. 3 Lewis Street Greenwich, CT United States.
- Devlin, J., Chang, M.-W., Lee, K., Google, K. T., and Language, A. I. (2019), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale.
- Huggingface (n.d.). "Transformers — transformers 4.0.0 documentation," *huggingface*, Available at <https://huggingface.co/transformers/>.
- Hutto, C. J. & G. (2014), "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.," *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Available at <https://github.com/cjhutto/vaderSentiment#installation>.

REFERENCIAS

Mourri, Y. B., Kaiser, L., and Shyu, E. (n.d.). “Natural Language Processing with Attention Models,” *Coursera*, Available at <https://www.coursera.org/learn/attention-models-in-nlp>.

Navas-Loro, M., Rodríguez-Doncel, V., Santana, I., and Sánchez, A. (2017), “Additional Information on the Spanish Corpus for Sentiment Analysis towards Brands,” *Springer, Cham*. https://doi.org/10.1007/978-3-319-66429-3_68.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018), *Improving Language Understanding by Generative Pre-Training*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020), *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, *Journal of Machine Learning Research*.

Stanford (2019), “Stanford CS224N: NLP with Deep Learning | Winter 2019 | Lecture 14 – Transformers and Self-Attention - YouTube,” *stanfordonline*, Available at https://www.youtube.com/watch?v=5vcj8kSwBCY&list=PLakWuueTN59e7ck3fB5lv_y_aHp_hUvMfgA&index=6&t=854s.

REFERENCIAS

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017), “Attention is all you need,” *arXiv*.

○ ¡GRACIAS!

¿Preguntas?