

## ***NLP***

### Long short term memory (LSTM)

Msc. Rodrigo Cardenas Szigety  
rodrigo.cardenas.sz@gmail.com

Esp. Ing. Hernán Contigiani  
hernan4790@gmail.com

# Programa de la materia



**Clase 1:** Introducción a NLP, Vectorización de documentos.

**Clase 2:** Preprocesamiento de texto, librerías de NLP y Rule-Based Bots.

**Clase 3:** Word Embeddings, CBOW y SkipGRAM, representación de oraciones.

**Clase 4:** Redes recurrentes (RNN), problemas de secuencia y estimación de próxima palabra.

**Clase 5:** Redes LSTM, análisis de sentimientos.

**Clase 6:** Modelos Seq2Seq, traductores y bots conversacionales.

**Clase 7:** Celdas con Attention. Transformers, BERT & ELMo, fine tuning.

**Clase 8:** Cierre del curso, NLP hoy y futuro, deploy.

\*Unidades con desafíos a presentar al finalizar el curso.

\*Último desafío y cierre del contenido práctico del curso.



*"Una celda RNN no puede mantener mucho el contexto o memoria (tienen problemas de short memory) pero se entrenan más rápido y son más baratas de ejecutar (tienen menos parámetros)"*

Es muy probable que una celda RNN tenga un buen desempeño en la primera sentencia de ejemplo, pero muy improbable en la segunda por la distancia entre la palabra clave y la palabra objetivo:

Ejemplo 1:

*"Vivo en **Argentina** desde muy pequeño, donde me enseñaron a hablar castellano"*

Ejemplo 2:

*"Vivo en **Argentina** desde muy pequeño, mis padres viajaron a Argentina en búsqueda de nuevas oportunidades. En la escuela me enseñaron a hablar muy bien castellano"*

# Long short term memory (LSTM)

[LINK](#)



*"Se introduce este tipo de celda neuronal con mayor persistencia de memoria para lograr capturar relaciones de palabras a largo plazo".*



Se crearon en 1997, se adoptó como la layer principal para problemas de secuencia en 2014 y en 2018 unieron fuerzas con los "attentions".



Desplazaron completamente a las redes RNN, ya que el costo adicional de las LSTM es marginal respecto al beneficio que otorgan



Se basan en el principio de ponderar la importancia de una palabra respecto al contexto futuro/pasado (key words).

¿Comprarías este producto?

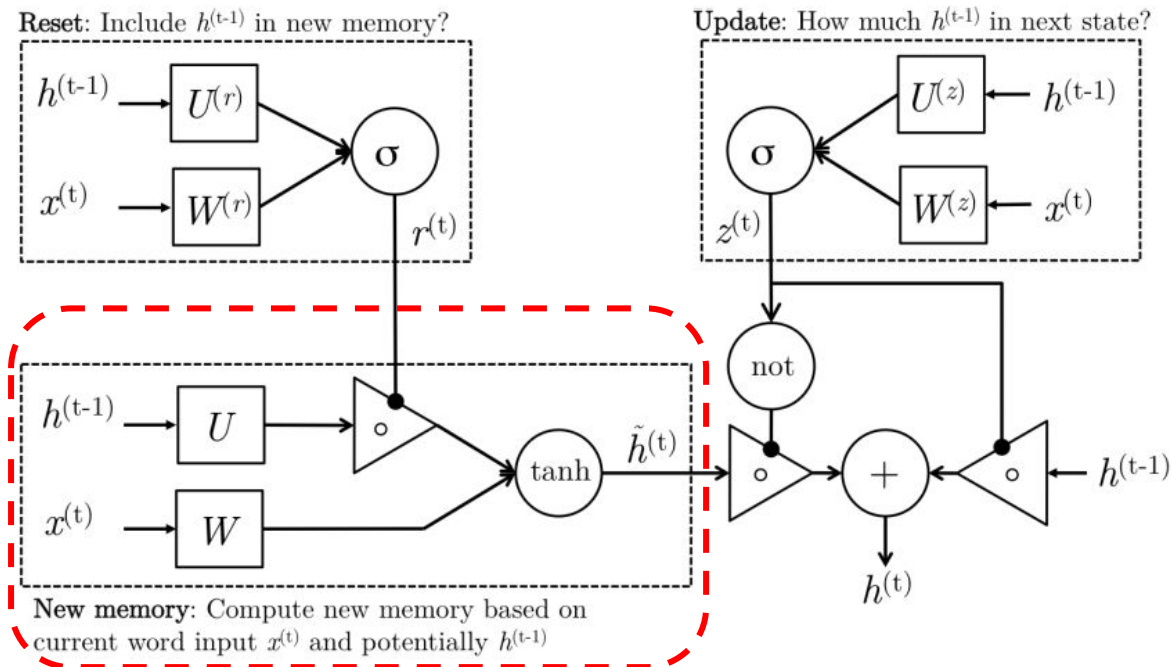
*"**Incredible!** El producto es lo que venden, hace lo que tiene que hacer y me **ayudó mucho** a resolver los problemas que tenía. Lo **volvería a comprar** sin dudas"*

# Gated Recurrent Units (GRU)

[LINK](#)



"Evolución de las RNN para superar problemas de "short-memory", versión reducida de una LSTM".



Nuevos pesos a entrenar

$$\begin{aligned} z_t &= \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \\ r_t &= \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \\ \tilde{h}_t &= \tanh(r_t \circ U h_{t-1} + W x_t) \\ h_t &= (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \end{aligned}$$

(Update gate)

(Reset gate)

(New memory)

(Hidden state)

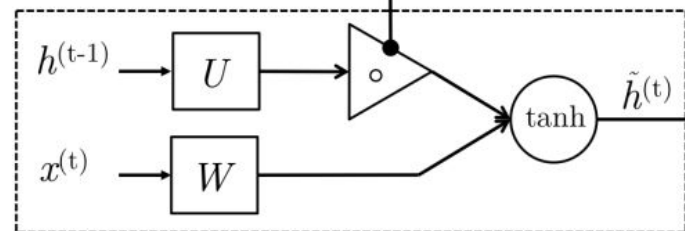
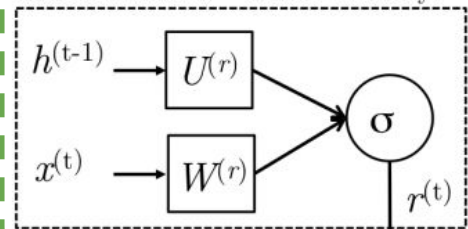
Cómo funciona la clásica RNN

# GRU - Reset Gate

[LINK](#)



Reset: Include  $h^{(t-1)}$  in new memory?



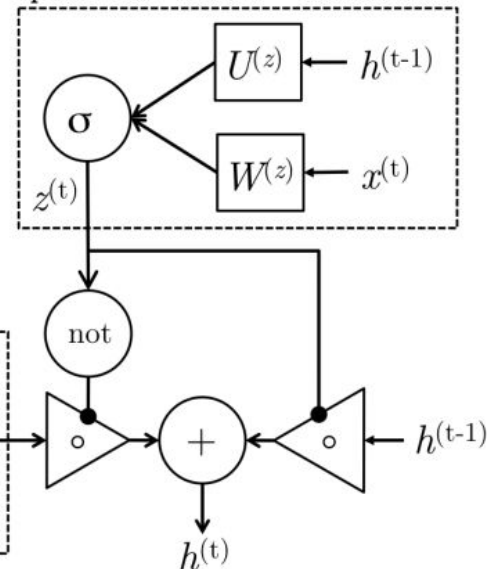
$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

$$\tilde{h}_t = \tanh(r_t \circ U h_{t-1} + W x_t)$$

$$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1}$$

Update: How much  $h^{(t-1)}$  in next state?



Determina qué tan importante es el evento pasado ( $h_{t-1}$ ) en la generación del nuevo estado de memoria ( $h_t^\wedge$ ).

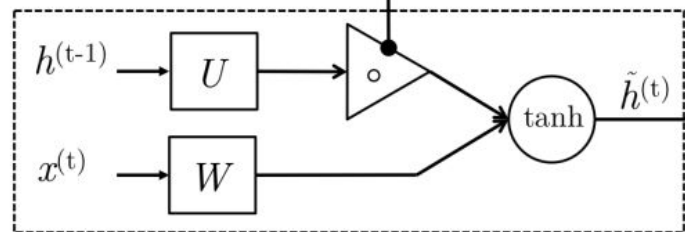
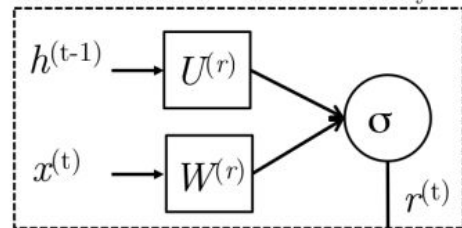
El sistema determina con una sigmoid el grado de (%) que el estado anterior se involucra en el nuevo, pudiendo determinar que este no fluye en absoluto (bypass)

# GRU - Update Gate

[LINK](#)



Reset: Include  $h^{(t-1)}$  in new memory?



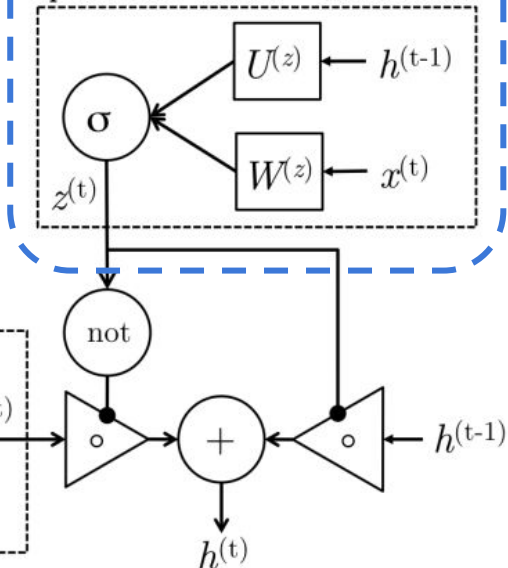
$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

$$\tilde{h}_t = \tanh(r_t \circ U h_{t-1} + W x_t)$$

$$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1}$$

Update: How much  $h^{(t-1)}$  in next state?

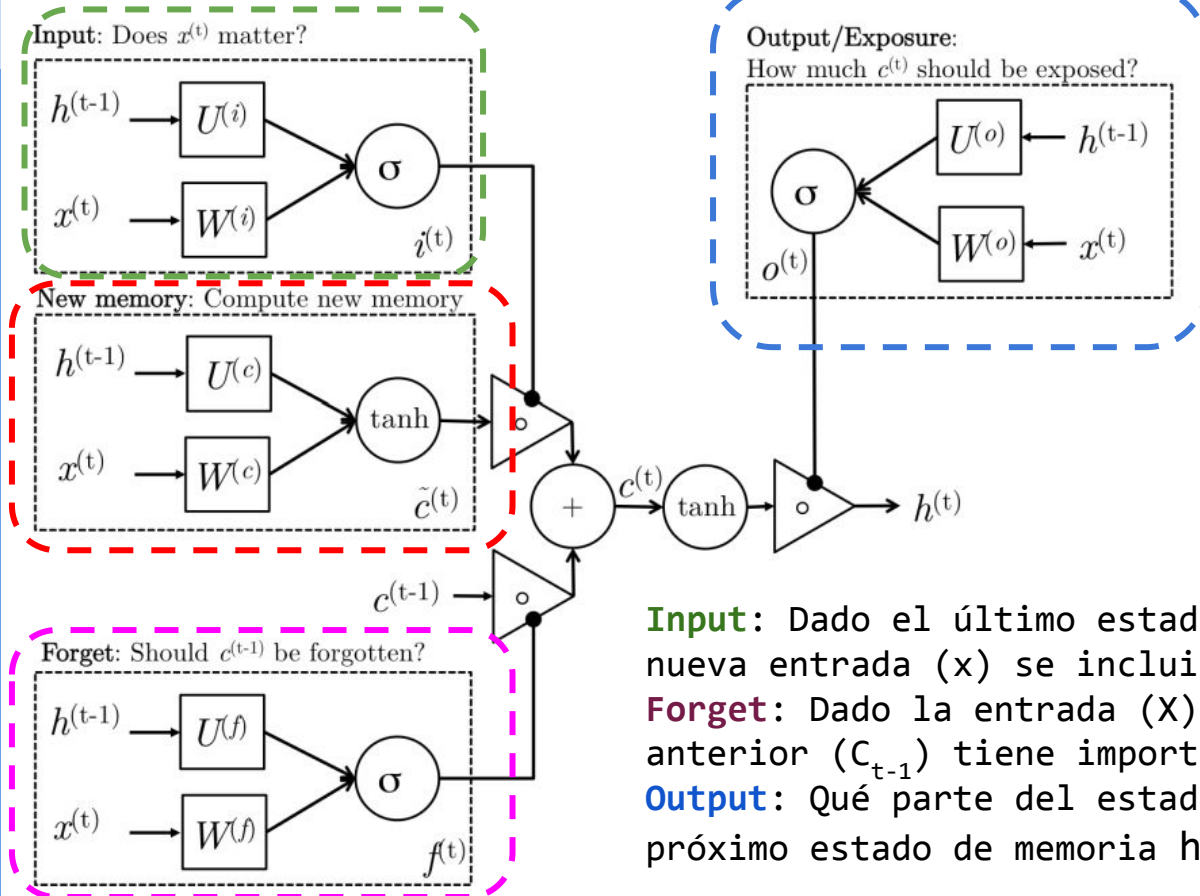


Determina qué tanto se involucra el estado anterior ( $h_{t-1}$ ) en la salida del sistema o siguiente estado ( $h_t$ ). Decide cuánta información nueva agregar o desechar al estado interno de la celda.

El sistema podría determinar que el estado actual no se propague al siguiente estado, y por lo tanto toda la información del estado anterior pasa (bypass) al siguiente.

# LSTM approach

[LINK](#)



$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1})$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1})$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1})$$

$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1})$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ \tanh(c_t)$$

**Input:** Dado el último estado ( $h_{t-1}$ ) evalúa cuánto de la nueva entrada ( $x$ ) se incluirá en la memoria ( $C_t$ ).

**Forget:** Dado la entrada ( $X$ ) cuanto del estado de memoria anterior ( $C_{t-1}$ ) tiene importancia en el nuevo estado.

**Output:** Qué parte del estado de memoria ( $C_t$ ) pasa al próximo estado de memoria  $h_t$ .



# ¿Cómo se refleja el concepto de la memoria?



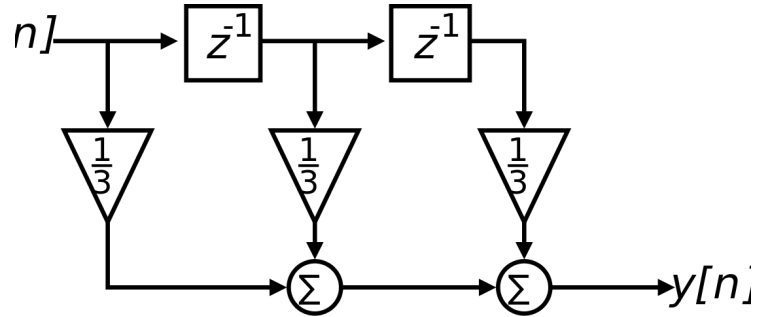
En este ejemplo se ve como funciona un filtro de media móvil por ventana, en donde el resultado depende de los valores anteriores

D:	10	-2	7	6	2	0
----	----	----	---	---	---	---

MA:	5	3.66	5	2.66
-----	---	------	---	------

$$MA_n = \frac{\sum_{i=1}^n D_i}{n}$$

Moving average



## ¿Problemas?

Todos los estados tienen el mismo peso o significancia  
Tiene problema de memoria corta (los valores salen de la ventana y se olvidan)

# ¿Cómo se refleja el concepto de gate?



En una red LSTM los valores anteriores están afectados por un coeficiente variable entrenado por la red.

V:	10	-2	7	6	2	0
----	----	----	---	---	---	---

W:	1	-1	0	2	0	5
----	---	----	---	---	---	---

M:	4	4.66	4	4
----	---	------	---	---

$$M = \frac{\sum_{t=1}^n W_t * V_t}{\sum_{t=1}^n W_t}$$

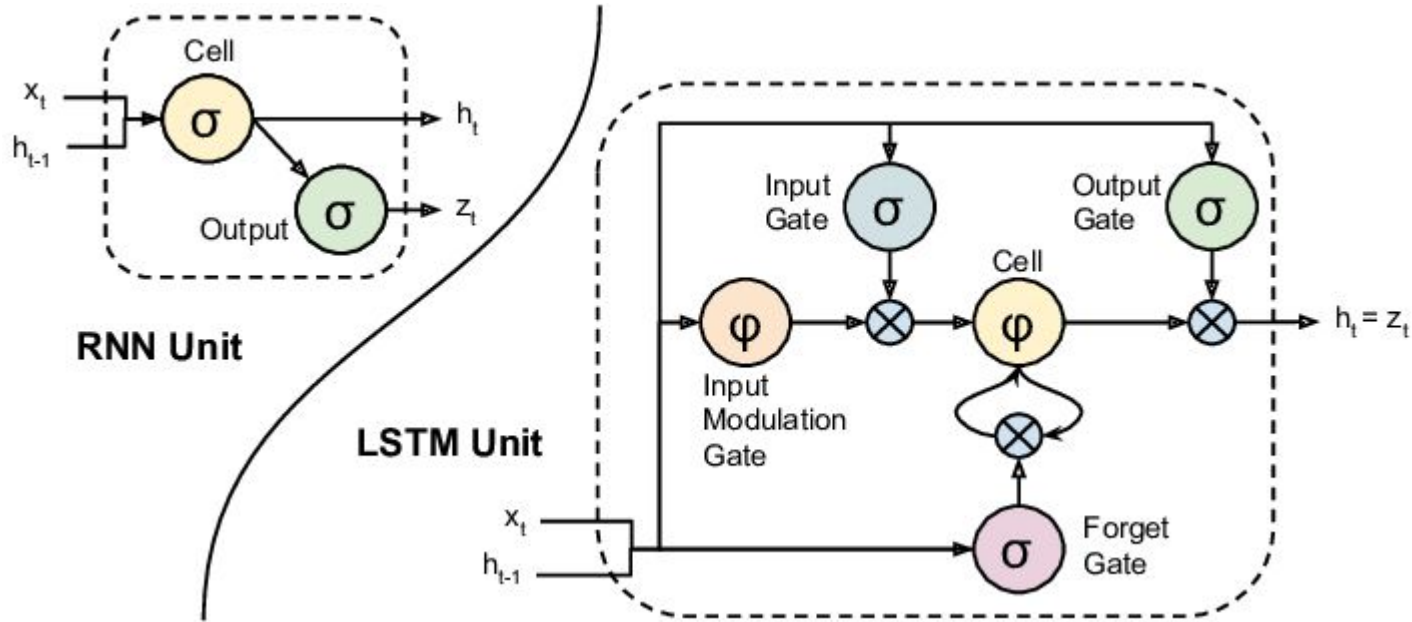
M = Average value  
V = Actual value  
W = Weighting factor  
n = Number of periods in the weighting group

Weighted Moving  
average

# LSTM vs RNN



Las celdas RNN son mucho más simples pero no puede abordar problemas más complejos como los que veremos de seq2seq

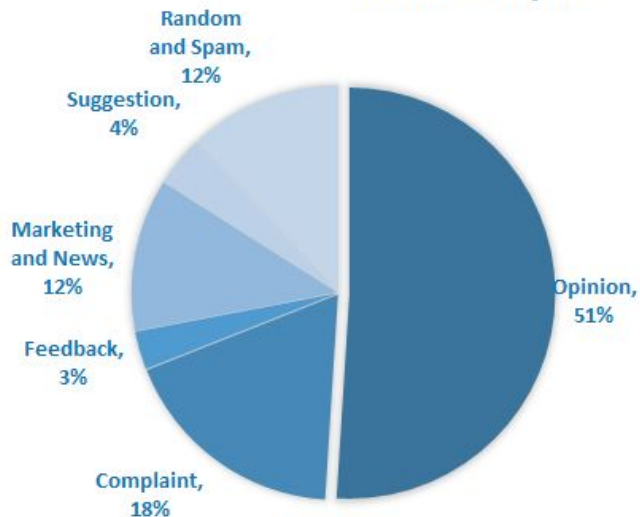


# Sentiment analysis



*"Es una forma de clasificar texto a fin de encontrar la intención o el sentimiento detrás de las palabras (positivo, neutral, negativo)"*

Intent Analysis



Sentiment Analysis



My experience  
so far has been  
fantastic!

POSITIVE



The product is  
ok I guess

NEUTRAL



Your support team is  
useless

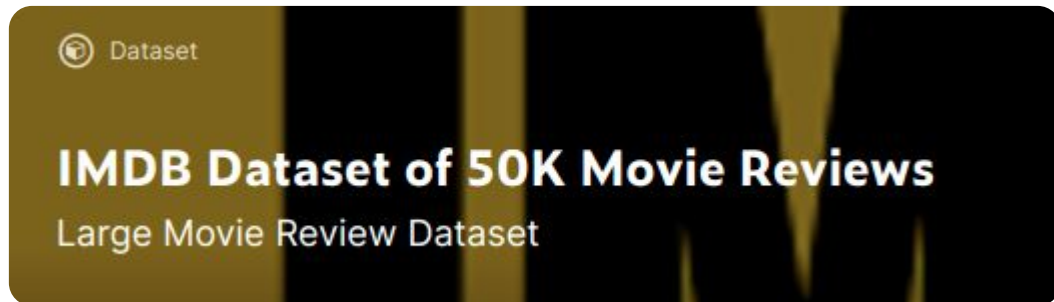
NEGATIVE

# IMDB dataset



Dataset con muchas críticas de películas en formato "positivo" o "negativo" (clasificación binaria de texto)

[LINK](#)



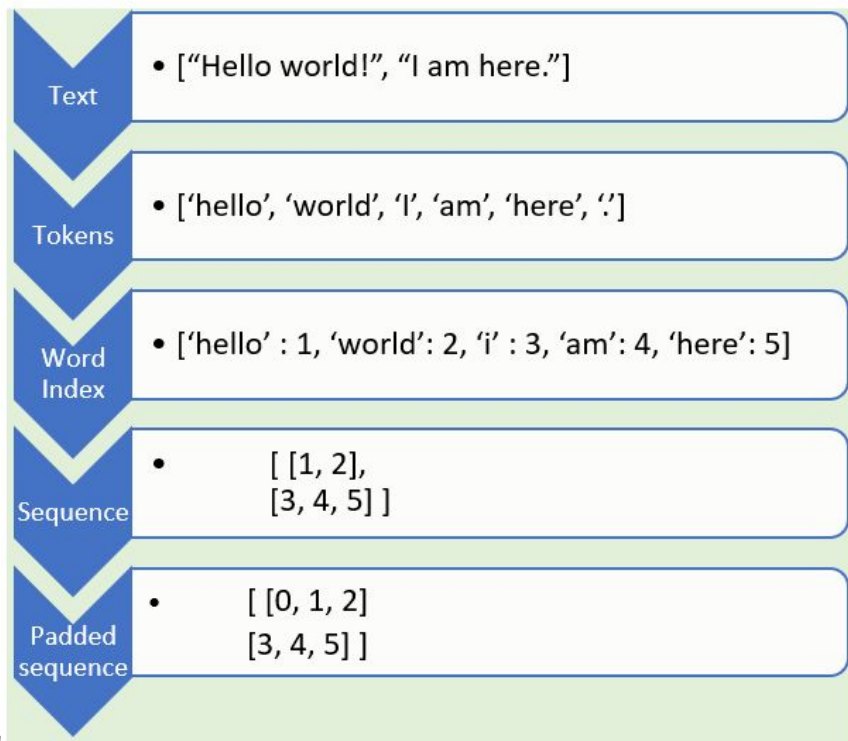
review	sentiment
49582 unique values	2 unique values
One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. The...	positive
Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his par...	negative

# Padding

[LINK](#)



Hoy trabajaremos con sentiment analysis el cual responde a la estructura many-to-one (text\_sequence to class/label)



Es necesario garantizar que la longitud de la secuencia de entrada siempre será del mismo largo, para eso se agregan ceros al comienzo o final de las cadenas de texto más cortas (padding)

```
padded_seq =  
[[ 2  6  3  7  8]  
 [ 0  2  9  3  4] <----- 0 Padded at the beginning  
 [ 0  0  0  5 10]  
 [ 0  0  0  5  4]]
```



Link al Colab



*LINK*



Link al Colab



*LINK*

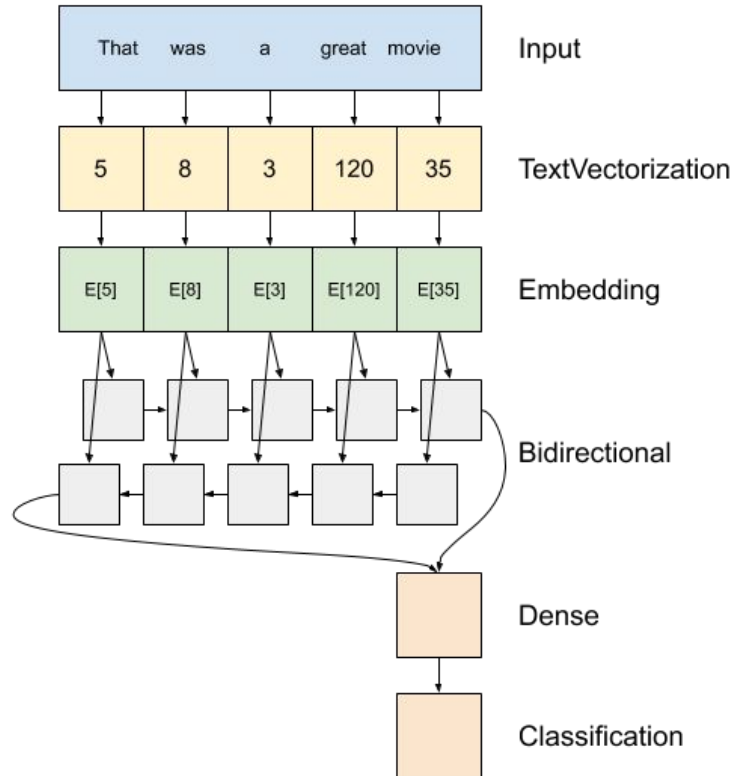


# Embeddings + LSTM + Classifier

[LINK](#)



Arquitectura de alto nivel de un modelo “sentiment analysis”



# Pre-trained Embedding layer

[LINK](#)

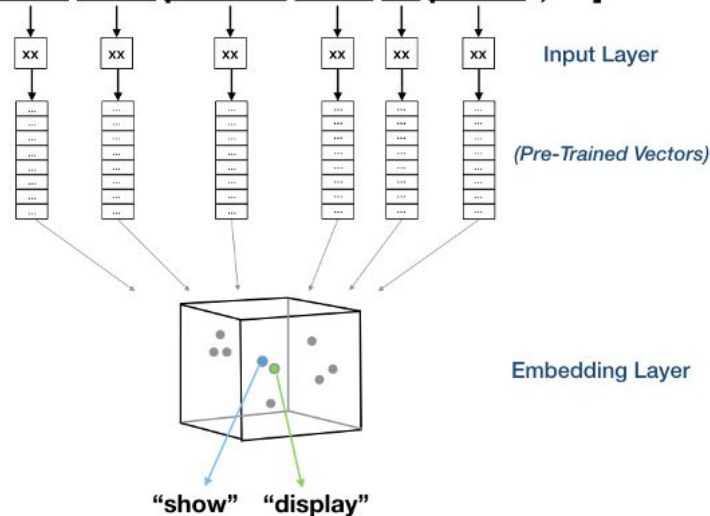
[LINK](#)



*"Utilizar embeddings pre-entrenados (GloVe / FastText) en la layer de Embeddings de Keras"*

```
Embedding(input_dim=vocab_size, # definido en el Tokenizador
          output_dim=embed_dim, # dimensión de los embeddings utilizados
          input_length=in_shape, # máxima sentencia de entrada
          weights=[embedding_matrix], # matrix de embeddings
          trainable=False)) # marcar como layer no entrenable
```

**[“I want to search for blood pressure result history”,  
“Show blood pressure result for patient”, ... ]**



a	1
am	2
as	3
act	4
all	5
...	6
...	7
...	8
...	9
...	10
...	11
...	12
...	...
...	100
...	...
...	1000
...	...
...	10000
...	...
...	...



Link al Colab



*LINK*



Utilizar Embeddings +  
LSTM para clasificar  
críticas de  
compradores de ropa



*LINK*

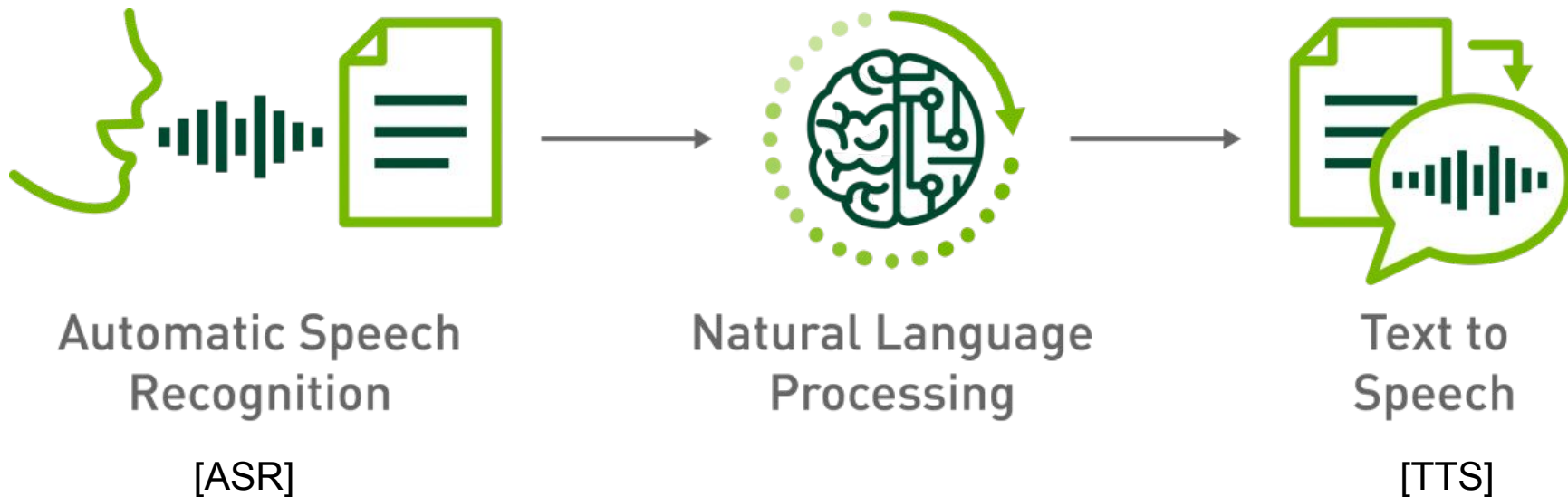
# Contenido extra

## Rápida observación de Speech processing

[LINK](#)



Proceso que permite transformar audio a texto (ARS) o texto a audio (TTS)



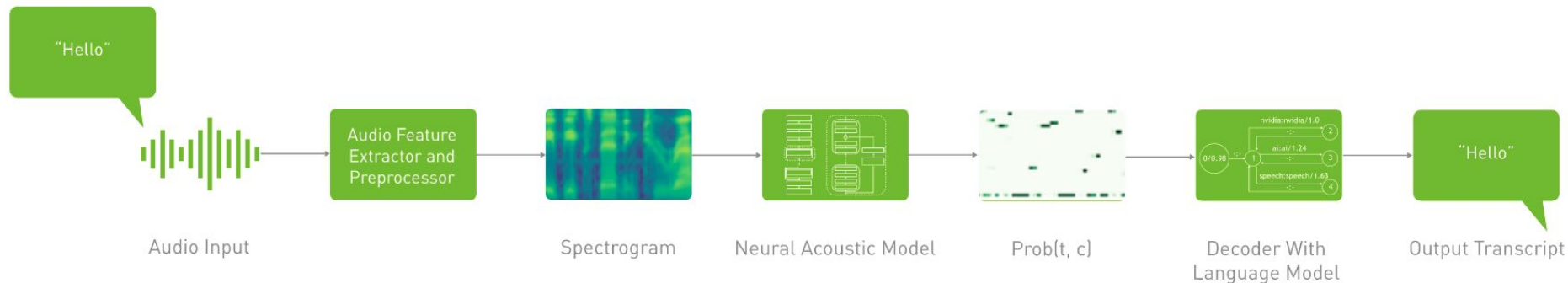
# Contenido extra

## Speech to text (ASR)

[LINK](#)



- Primer proceso es eliminar o ignorar el ruido (filtros)
- Transformar el audio a un espectrograma para obtener features.
- Transformar los features a posibles palabras con un modelo neuronal acústico.
- Utilizar un modelo de NLP para transformar las palabras reconocidas en una sentencia/oración con significado.





Link al Colab



[LINK](#)



# ¡Muchas gracias!