NLP

Vectorización de texto

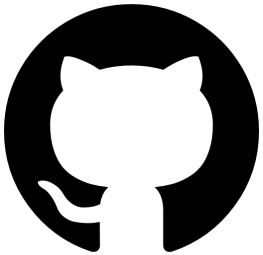
Dr. Rodrigo Cardenas Szigety
rodrigo.cardenas.sz@gmail.com

Programa de la materia

Clase 1: Introducción a NLP, Vectorización de documentos.
Clase 2: Word embeddings.
Clase 3: Redes recurrentes: Elman, LSTM y GRU.
Clase 4: Generación de secuencias.
Clase 5: CNNs, introducción a atención. Modelos de clasificación.
Clase 6: Modelos Seq2seq.
Clase 7: Mecanismo de atención, Transformers.
Clase 8: Grandes modelos de lenguaje.

*Unidades con desafíos a presentar al finalizar el curso.
*Último desafío y cierre del contenido práctico del curso.

Link Github de la materia



https://github.com/FIUBA-Posgrado-Inteligencia-Artificial/procesamiento_lenguaje_natural

En el Github van a encontrar...

LINK

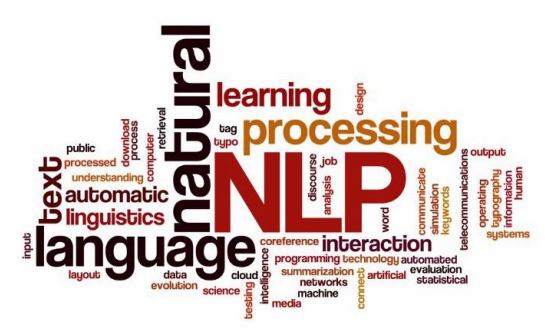
Trabajaremos en la clase con Keras/Tensorflow.

No obstante, pueden usar el framework que más cómodo les resulte

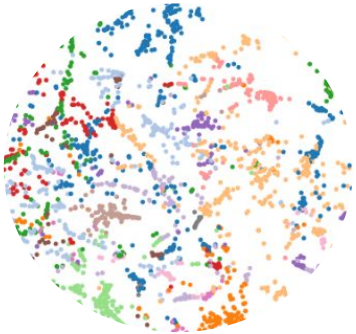
- Creado por Google
- Utilizado principalmente en la industria y en el despliegue.
- Los bloques del framework son bastante cerrados.
- Posee muchas librerías y tools que de ayudan.
- Muchas tools para despliegue y debugging

- Creado por Facebook
- Utilizado principalmente en el campo académico e investigación.
- Los bloques del framework son totalmente abiertos.
- Posee pocas librerías o tools, hay que desarrollar mucho uno mismo.
- Los nuevos modelos de NLP salen antes en Pytorch que en Tensorflow

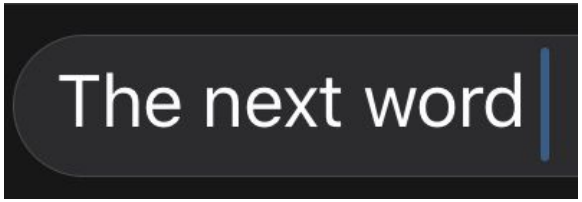
Desafíos semanales



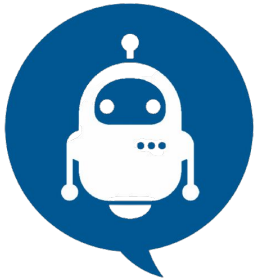
Vectorización de texto



Word embeddings



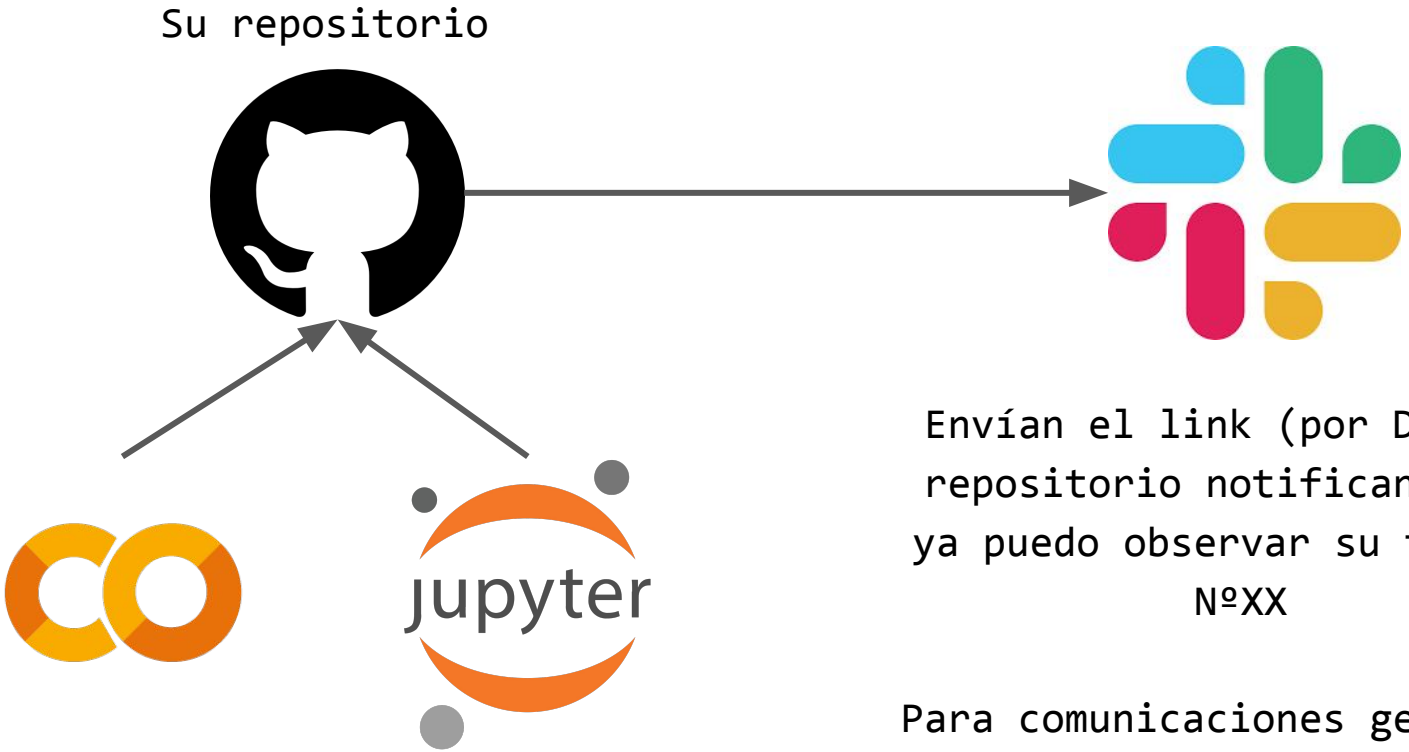
Generación de secuencias



Modelos seq2seq

Modelos de clasificación

¿Cómo me acercaran sus soluciones?



Envían el link (por DM) del repositorio notificando que ya puedo observar su trabajo NºXX

Para comunicaciones generales usamos el channel de Slack #nlp del workspace de CEIA

¿Cómo se evaluarán los desafíos?

	Clases								Recu
	1~2	2~3	3~4	4~5	5~6	6~7	7~8	8	
Desafío 1	9-10	9-10	8-9	8-9	7-8	7-8	6-7	6-7	4-6
Desafío 2		9-10	9-10	8-9	8-9	7-8	7-8	6-7	4-6
Desafío 3			9-10	9-10	8-9	7-8	7-8	6-7	4-6
Desafío 4				9-10	9-10	8-9	7-8	6-7	4-6
Desafío 5					9-10	9-10	8-9	7-8	4-6
						9-10	8-9	7-8	4-6
Desafío 6							9-10	8-9	4-6

PARA APROBAR EL CURSO TODOS LOS DESAFÍOS DEBEN SER ENTREGADOS Y EVALUADOS SATISFACTORIAMENTE

*La instancia de recuperación comienza luego de la última clase. La instancia de recuperación tiene una duración de una semana límite para terminar de entregar los desafíos.

¿Qué es NLP?

El procesamiento de lenguaje natural (PLN o NLP) es una disciplina que combina la **computación**, la **inteligencia artificial** y la **lingüística**, que estudia métodos computacionales para interpretar el lenguaje humano.

El lenguaje:

Es cultural.

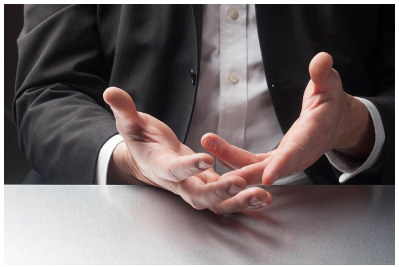
Es cambiante.

Es multimodal.

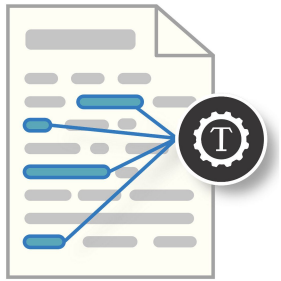
Es dependiente de múltiples contextos.

Modalidades del lenguaje

Señas, expresiones, contacto físico

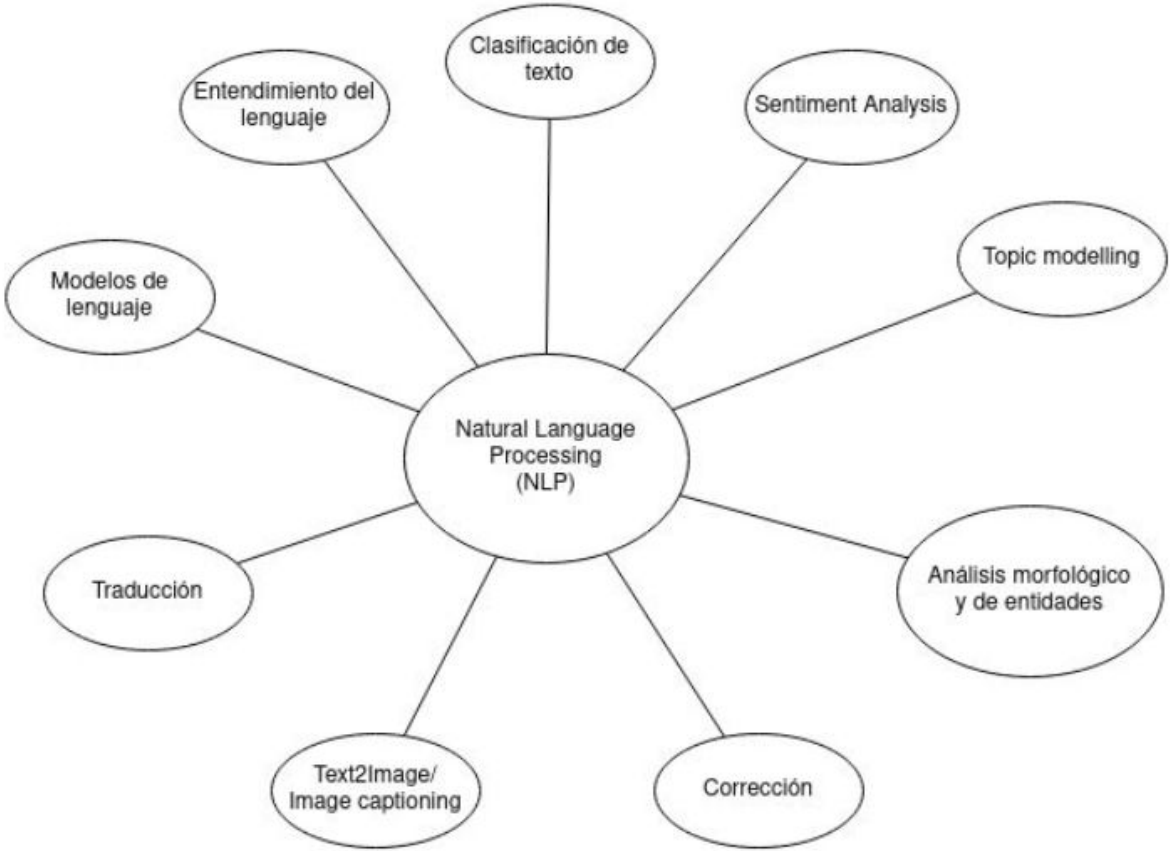


Oral

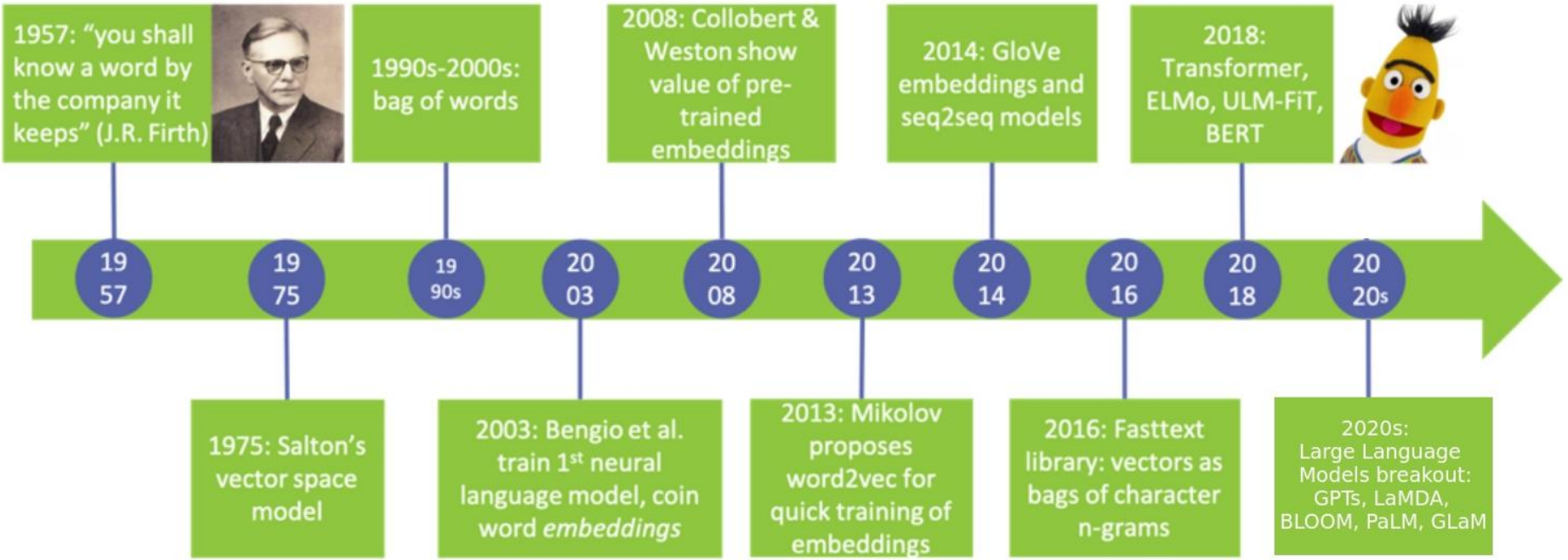


Texto

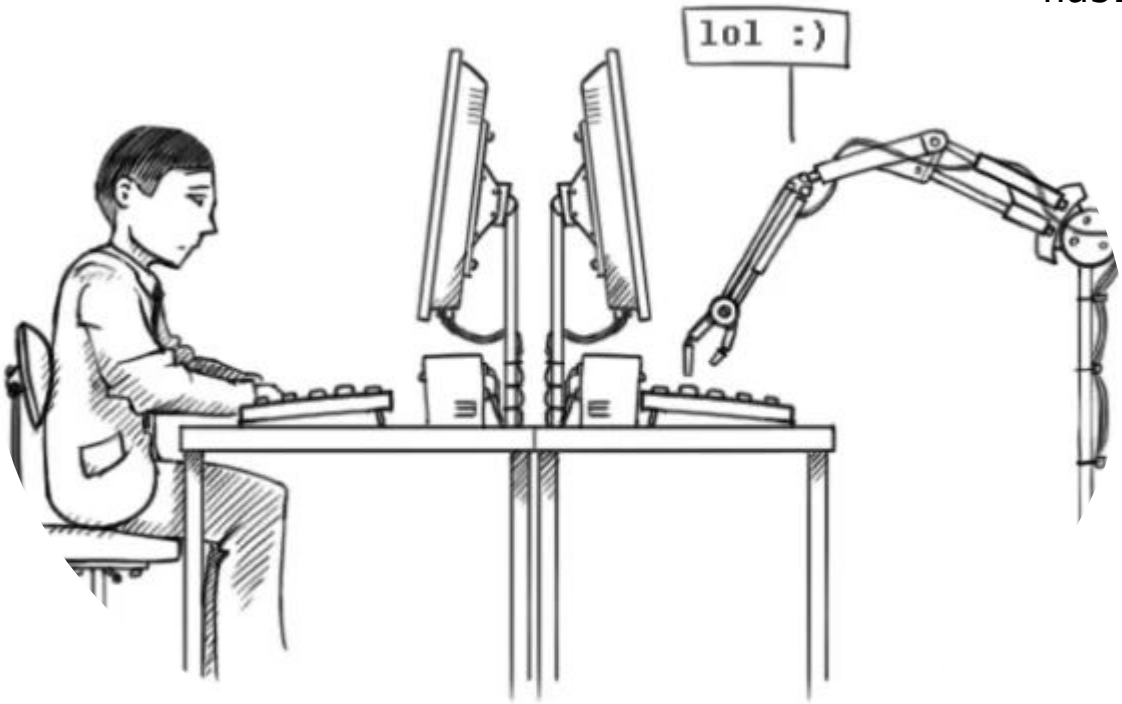
Problemas de NLP



Timeline


1957: "you shall know a word by the company it keeps" (J.R. Firth)
1990s-2000s: bag of words
2008: Collobert & Weston show value of pre-trained embeddings
2014: GloVe embeddings and seq2seq models
2018: Transformer, ELMo, ULM-FiT, BERT
19 57
19 75
19 90s
20 03
20 08
20 13
20 14
20 16
20 18
20 20s
1975: Salton's vector space model
2003: Bengio et al. train 1st neural language model, coin word *embeddings*
2013: Mikolov proposes word2vec for quick training of embeddings
2016: Fasttext library: vectors as bags of character n-grams
2020s: Large Language Models breakout: GPTs, LaMDA, BLOOM, PaLM, GLaM

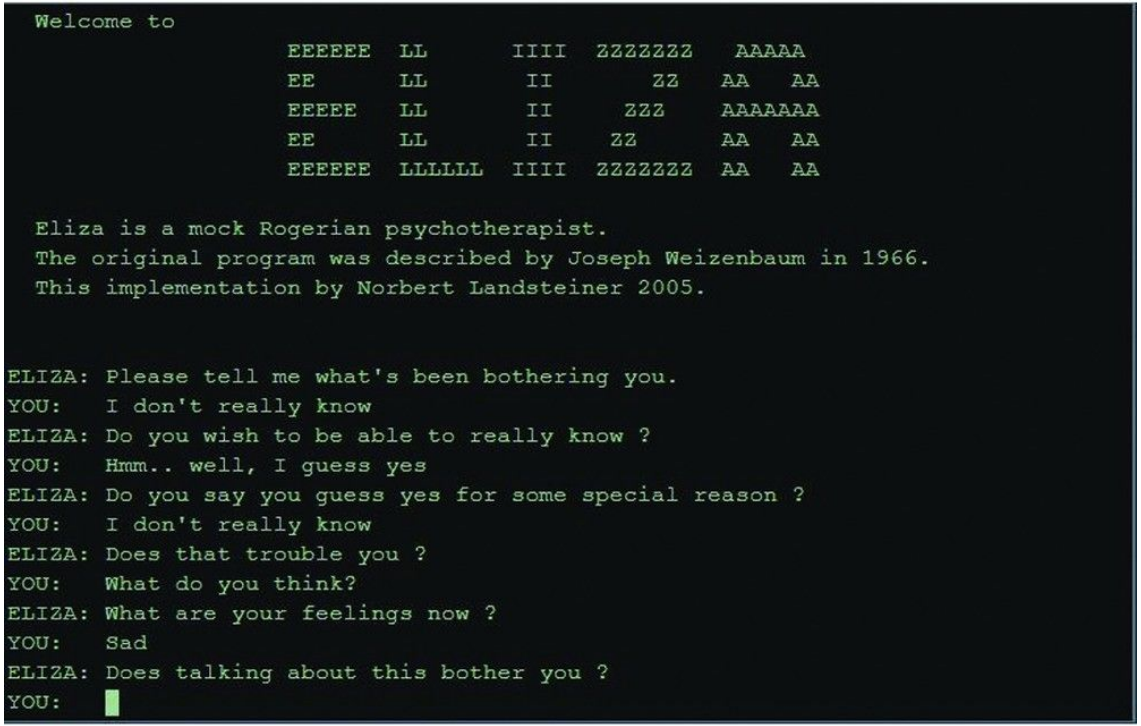
Test de turing (1950)



¿Puede una máquina hablar como si fuera un humano?

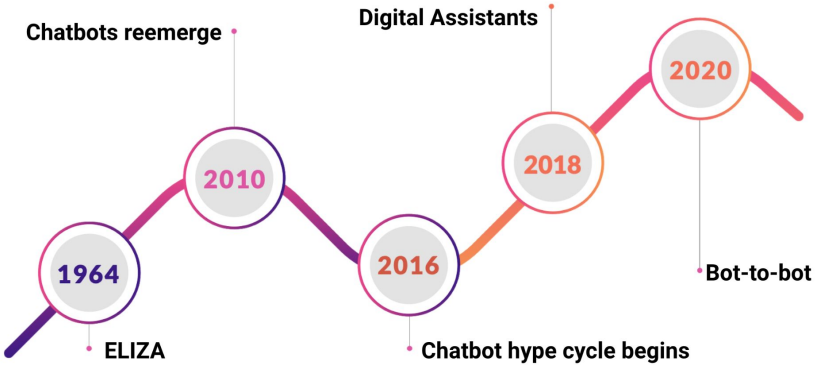
¿Superado?
(LaMDA, ChatGPT)

Eliza (1964) - Primer bot NLP



Se basaba en encontrar las palabras claves del texto y utilizarlas para formular una pregunta

Los chatbots y asistentes virtuales...



IBM Watson
2006

Apple Siri
2010

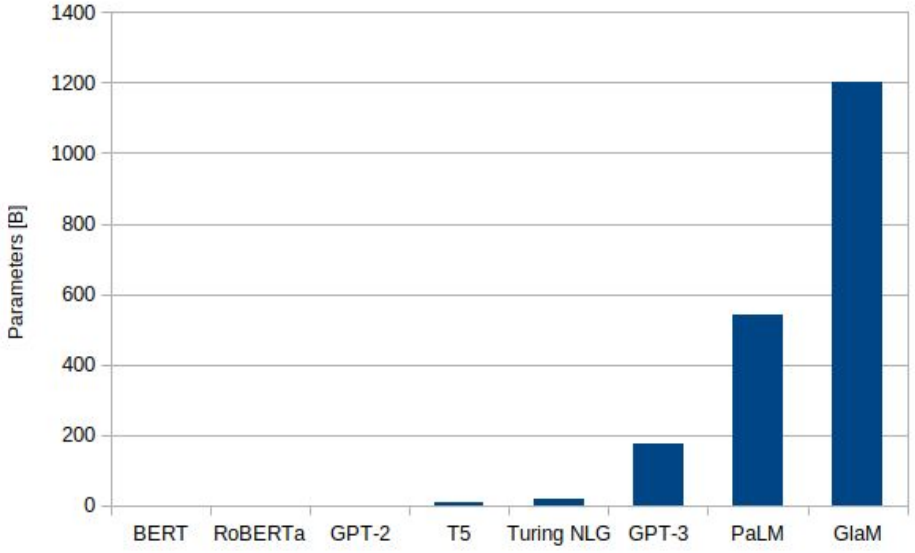
Google Now
2012

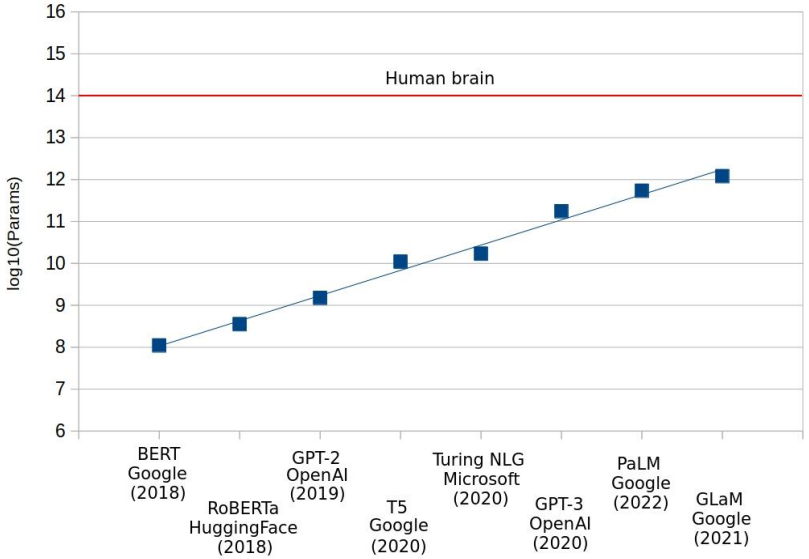
Amazon
alexa
2015

Microsoft
Cortana
2015

Huawei
Celia
2020

Los modelos que transformaron NLP

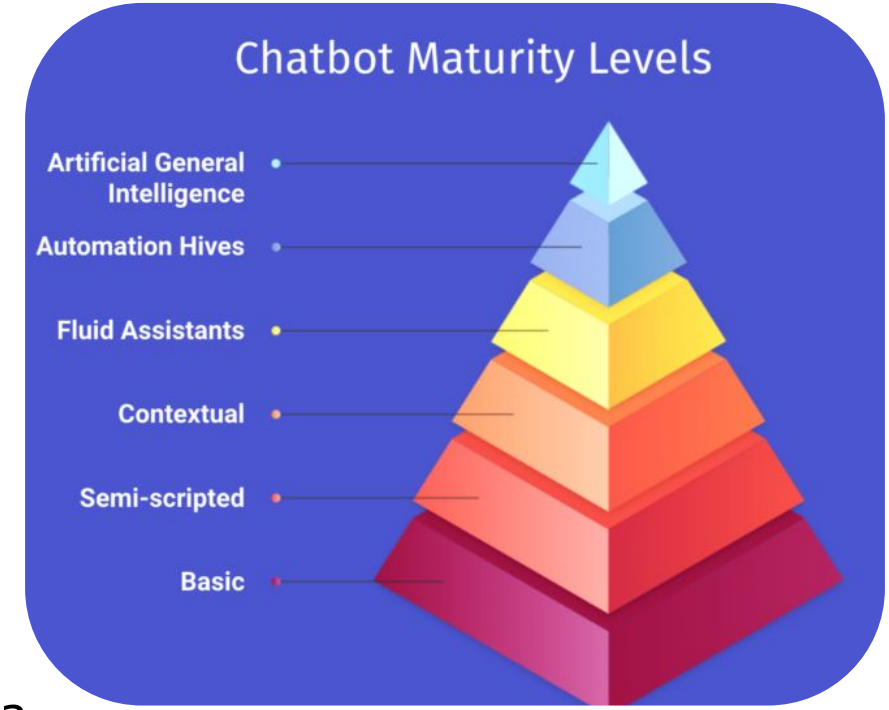




model.fit() de GPT-3 se estima en 12M U$S

800 GB

Elegir la herramienta que más se ajusta a sus problemas

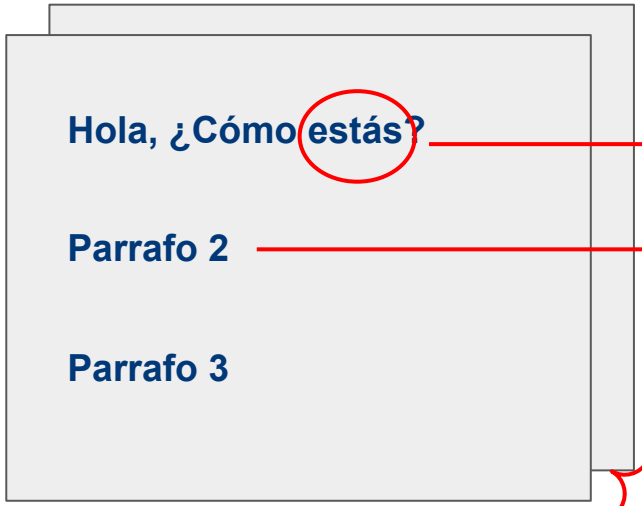


inteligencia

costo

Vectorización de texto

LINK GLOSARIO



Término t: palabra/símbolo "t" del documento

Document: su largo es variable, normalmente una sentencia/oración/párrafo.

Corpus: conjunto de documentos, forman todo el vocabulario.

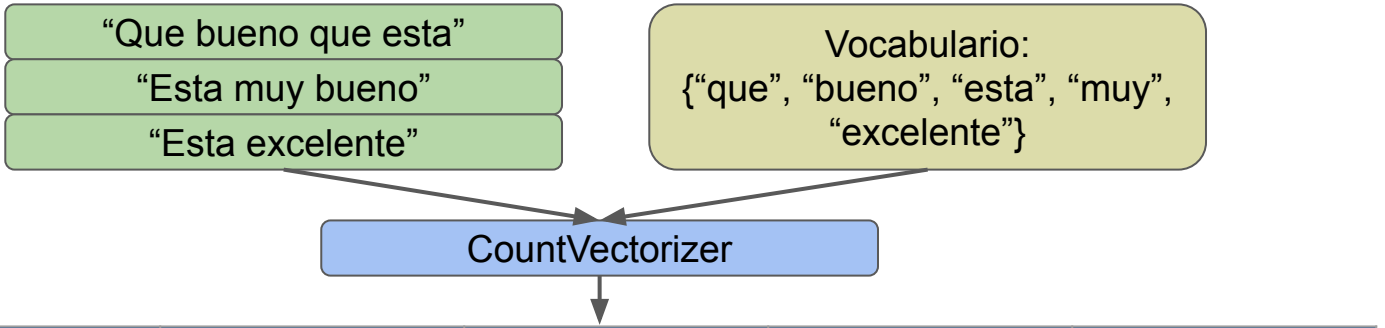
No podemos ingresar texto a una red
¿Cómo transformamos palabras a números?

vectorización →

Vectores de palabras/documentos

Vectores de frecuencia/conteo

"Por cada documento en el corpus se calcula un vector que representa cuántas veces cada palabra del vocabulario aparece en ese documento"

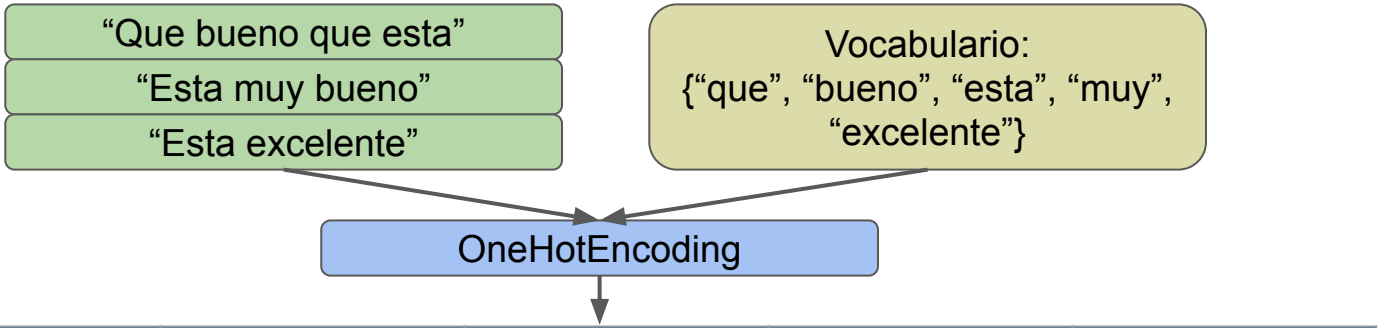


que	bueno	esta	muy	excelente
2	1	1	0	0
0	1	1	1	0
0	0	1	0	1

Los vectores tienen el tamaño del vocabulario

Vectores One-hot encoding (OHE)

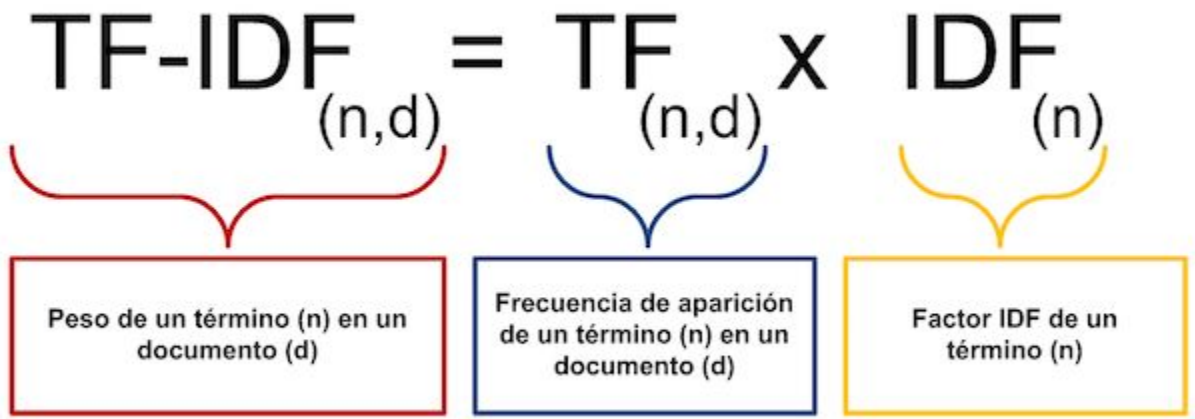
"Por cada documento en el corpus se calcula un vector que representa si cada palabra del vocabulario aparece o no en ese documento"



que	bueno	esta	muy	excelente
1	1	1	0	0
0	1	1	1	0
0	0	1	0	1

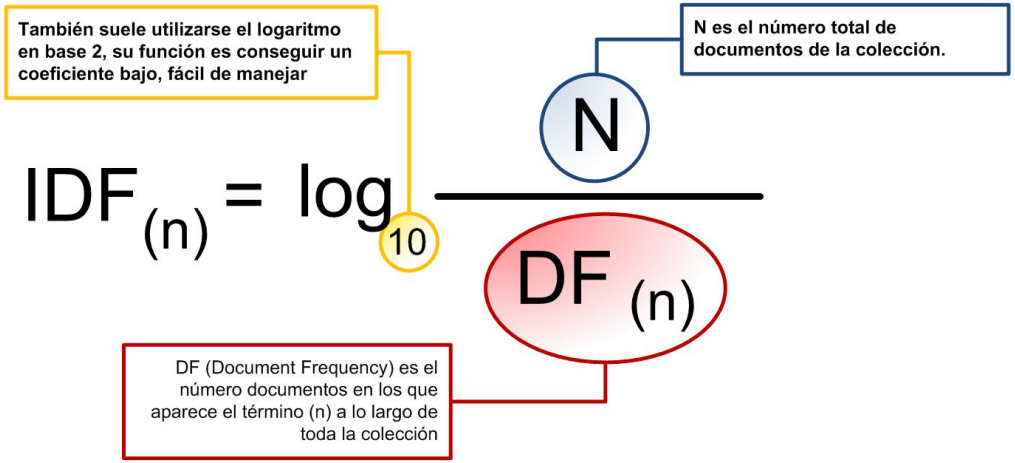
TF-IDF (Term frequency-Inverse term frequency)

"Se utiliza como indicador de cuán importante es una palabra (término) en un documento"



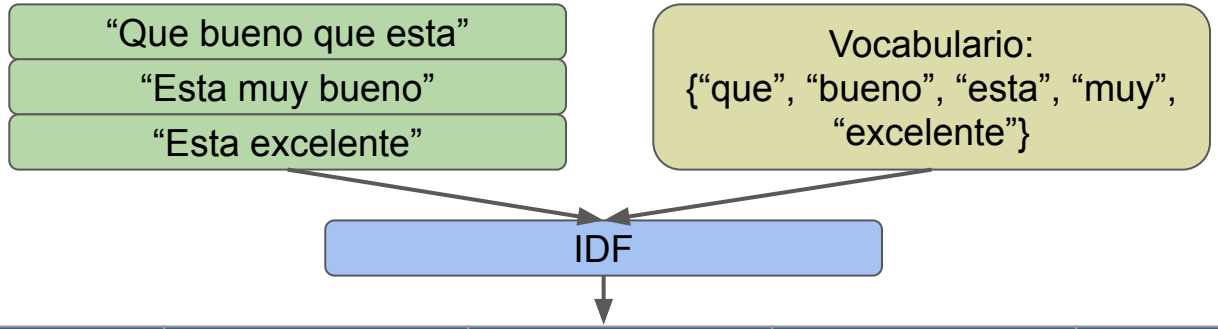
Vector IDF (Inverse Document Frequency)

"Proporción de documentos en el corpus que poseen el término"



Si el término aparece en todos los documentos el IDF será cero (es popular y por lo tanto aporta poco valor)

Vector IDF



que	bueno	esta	muy	excelente
log(3/1)	log(3/2)	log(3/3)	log(3/1)	log(3/1)

que	bueno	esta	muy	excelente
0.477	0.176	0	0.477	0.477

Se obtiene como la división de la cantidad de documentos sobre la suma en axis=0 (vertical) del OneHotEncoding.

Vector TF-IDF

"Que bueno que esta"
"Esta muy bueno"
"Esta excelente"

Vocabulario:
{"que", "bueno", "esta", "muy", "excelente"}

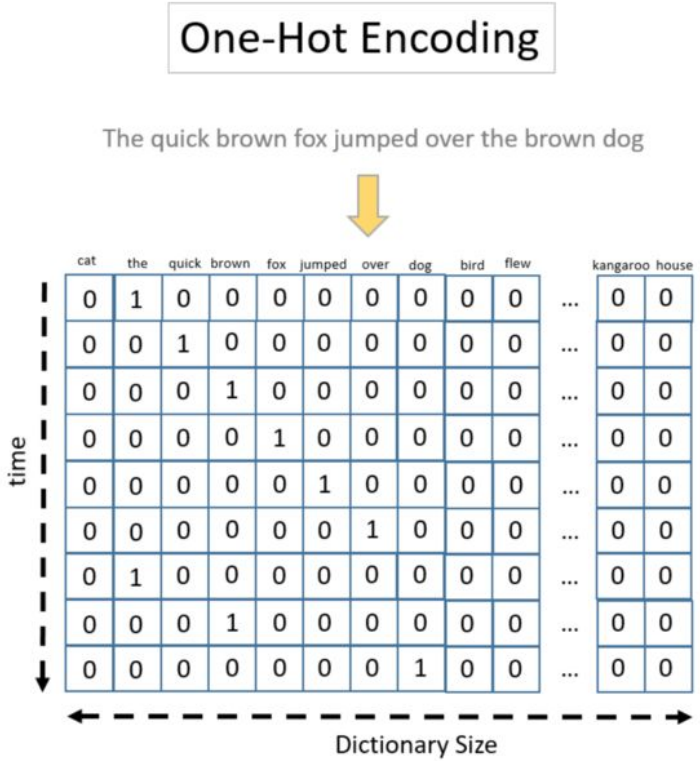
IDF

que	bueno	esta	muy	excelente
log(3/1)	log(3/2)	log(3/3)	log(3/1)	log(3/1)

TF-IDF

que	bueno	esta	muy	excelente
2 * log(3/1)	1 * log(3/2)	1 * log(3/3)	0 * log(3/1)	0 * log(3/1)
0 * log(3/1)	1 * log(3/2)	1 * log(3/3)	1 * log(3/1)	0 * log(3/1)
0 * log(3/1)	0 * log(3/2)	1 * log(3/3)	0 * log(3/1)	1 * log(3/1)

Esparsidad de los vectores de conteos (Frecuencia/OHE/TF-IDF)



¡El idioma inglés tiene más de 180.000 palabras en su vocabulario en uso!

¡La representación es sumamente esparsa!

No estamos aprovechando eficientemente la dimensionalidad del espacio de vectores.

Representación de documentos y palabras

Similitud coseno

"Se utiliza para evaluar la dirección de dos vectores"

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Similitud coseno = 1 → los vectores tienen la misma dirección.
Similitud coseno = 0 → los vectores son ortogonales.
Similitud coseno = -1 → los vectores apuntan en sentido contrario.

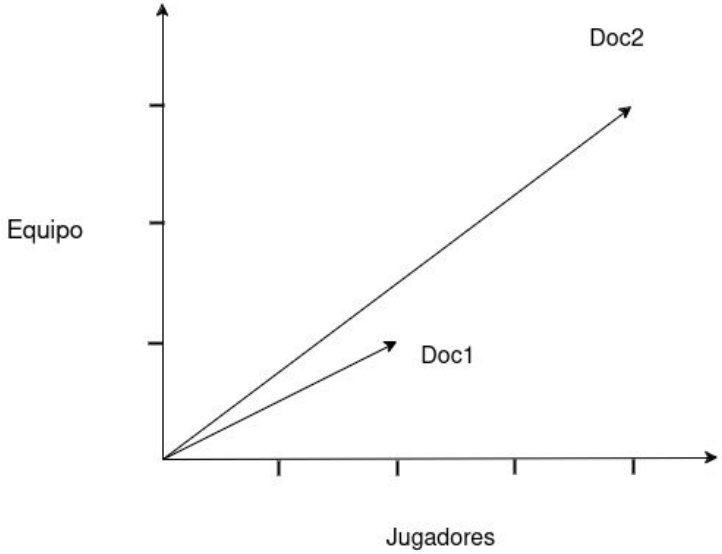
Intuición de la similitud coseno

Doc1

"Cada equipo en el campo tiene hasta once jugadores…"

Doc2

"... el equipo Argentino presentó a todos sus jugadores titulares… "



Para la distancia euclidea, los documentos son muy distintos. Para la similitud coseno, son similares.

Modelos de clasificación en NLP

ME FALTA INCLUIR N GRAMAS

BIBLIO DE REFERENCIA DAN JURAFSKY

Modelo de clasificación Naïve Bayes

Se tiene un vocabulario de tamaño V y un corpus anotado de N documentos que se pueden clasificar en C clases. Cada documento se representa como un vector [x0,...,xV-1].

Teorema de Bayes

$$\underbrace{P(c_i|[x_0,\ldots,x_{V-1}])}_{\text{Implementa un modelo probabilístico de clasificación}} = \frac{P([x_0,\ldots,x_{V-1}]|c_i)P(c_i)}{\boxed{P([x_0,\ldots,x_{V-1}])}}$$

Es un factor cte.

$$P(c_i|[x_0,\ldots,x_{V-1}]) \propto \underbrace{P([x_0,\ldots,x_{V-1}]|c_i)}_{\text{Verosimilitud de los datos}} P(c_i)$$

Probabilidad a priori de cada clase.

Modelo de clasificación Naïve Bayes

Probabilidad a priori de cada clase.

$$P(c_i) = \frac{N_{c_i}}{N}$$

Hipótesis "Naïve"

$$P([x_0, \ldots, x_{V-1}]|c_i) = \prod_j^{V-1} P(x_j|c_i)$$

Sólo hay que calcular la verosimilitud de palabras por separado dada la clase.

Modelo multinomial

$$P(x_j|c_i) = p_{palabra\ j|c_i}^{x_j}$$

¡Bueno, bonito, barato!

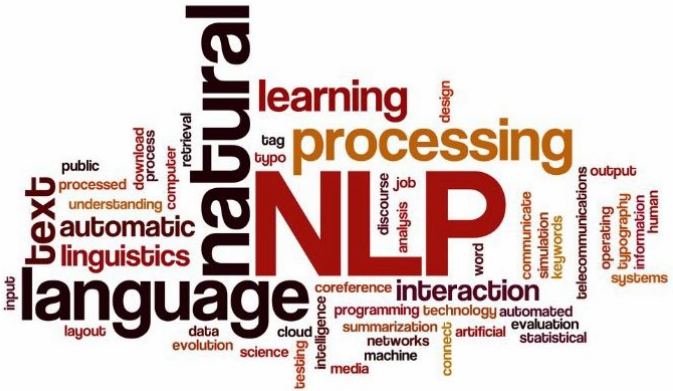
Explicación por sklearn

Modelo de clasificación Naïve Bayes

Explicación por sklearn

¡Bueno, bonito, barato!

Representaciones y modelos de tipo “Bolsa de palabras” (BOW)



Cualquier modelo o representación (vectorización) de texto en donde el resultado no se modifique con el órden de las palabras es un modelo o representación BOW.

CountVector, OHE, TF-IDF son ejemplos de representaciones BOW

Naïve Bayes es un ejemplo de clasificador tipo BOW

Desafio

Link al Colab

LINK

Sobre el uso de LLMs y asistentes de código en la materia...

¡¡Totalmente permitidos!! Se alienta a que los usen para lo que quieran (¡con criterio!).

Especificar modelo/asistente usado, fecha y prompts utilizados.

Explicar algunos hyperparams del tokenizador y el modelo (laplacian smoothing, n-grams?)
Meter pipeline, tokenización y algo de prepro

Practiquemos lo visto hasta ahora

Link al Colab

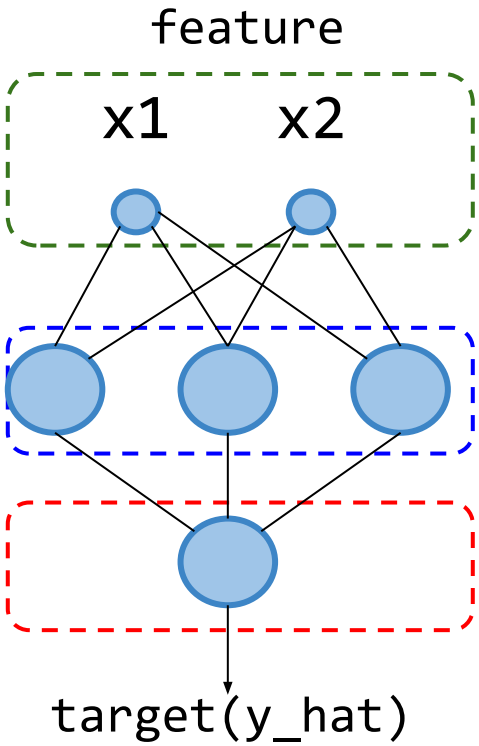
LINK

Keras Clasificación

Link al Colab

LINK

Keras & Tensorflow



input

Hidden

Output

```
# Crear un modelo secuencial
model = Sequential()

# Crear la capa de entrada y la capa oculta (hidden):
# --> tantas entradas (input_shape) como columnas de entrada
# --> tantas neuronas (units) como deseemos
# --> utilizamos "sigmoid" como capa de activación
model.add(Dense(units=3, activation='relu', input_shape=(2,)))

# Crear la output, tendrá tantas neuronas como salidas deseadas
model.add(Dense(units=1, activation='sigmoid'))
```