

NLP

Vectorización de texto

Docentes:

Dr. Rodrigo Cardenas Szigety

Dr. Nicolás Vattuone

emails: `rodrigo.cardenas.sz@gmail.com`

`nicolas.vattuone@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: Modelos de lenguaje y generación de secuencias.

Clase 5: CNNs, introducción a atención. Modelos de clasificación.

Clase 6: Modelos Seq2seq.

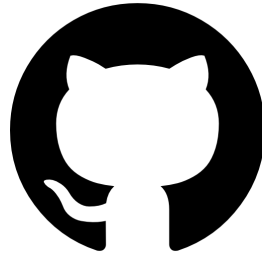
Clase 7: Mecanismo de atención, Transformers.

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.

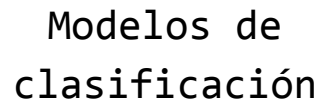
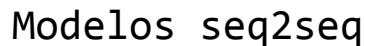
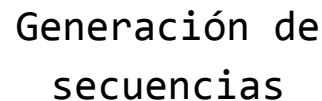
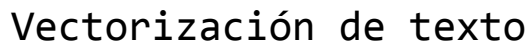
No obstante, pueden usar el framework que más cómodo les resulte



- 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



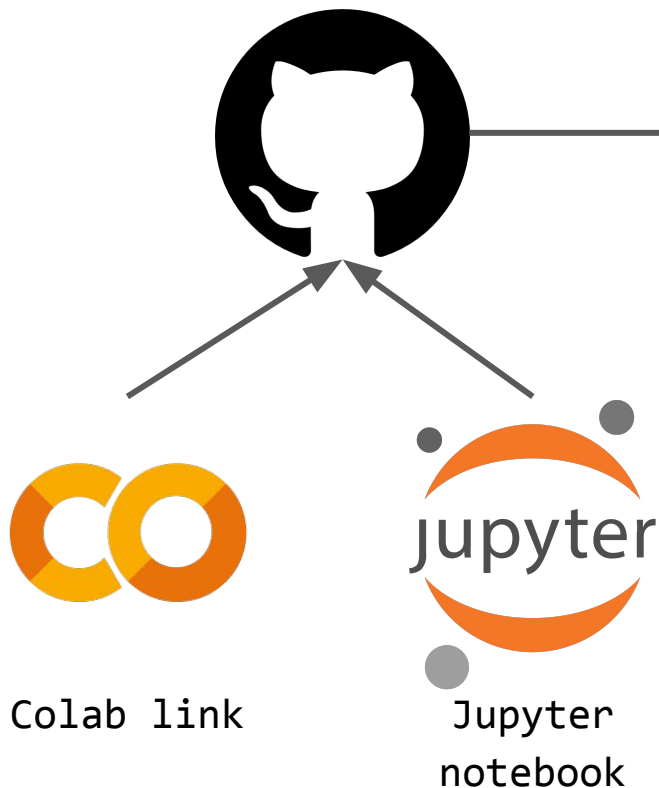
- Creado por Facebook
- Utilizado principalmente en el campo académico e investigación.
- Los bloques del framework son totalmente abiertos.
- Posee pocas librerías o tools, hay que desarrollar mucho uno mismo.
- Los nuevos modelos de NLP salen antes en Pytorch que en Tensorflow



¿Cómo me acercaran sus soluciones?



Su repositorio



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 ambiguo.

**“Los límites de mi lenguaje son los
límites de mi mundo”**

Ludwig Wittgenstein

Modalidades del lenguaje



Señas, expresiones, contacto físico



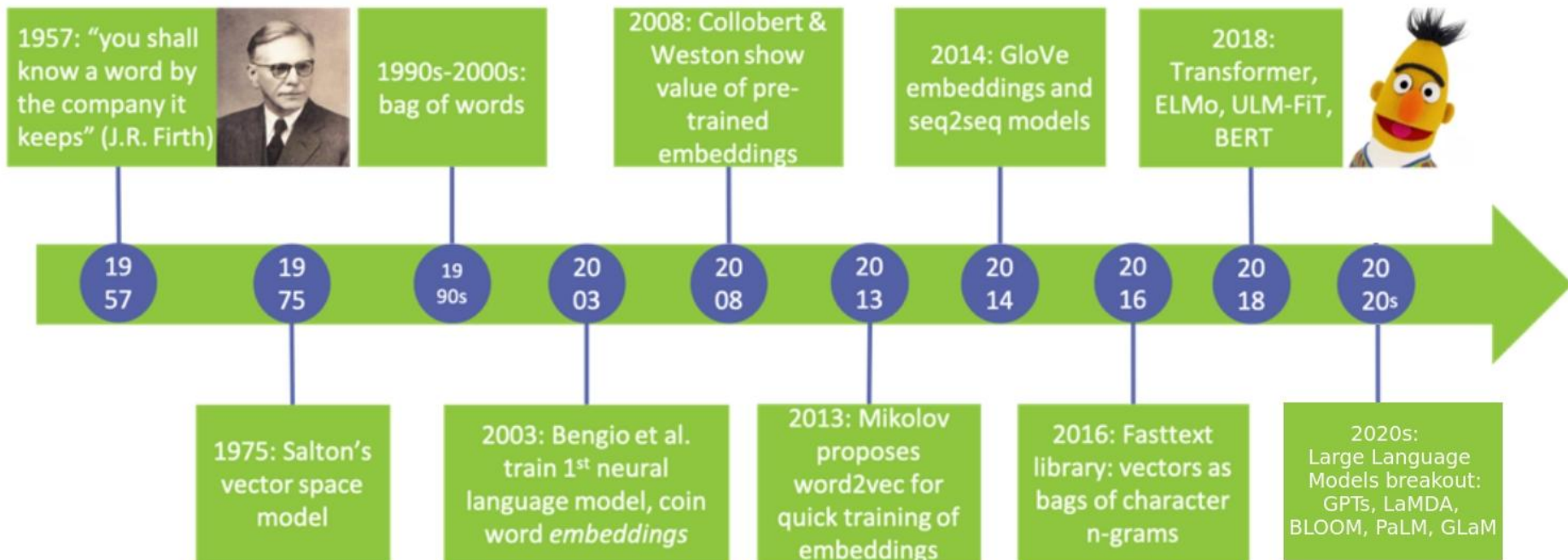
Oral

“Sin el lenguaje, el pensamiento es una
nebulosa vaga e inexplorada” -
Ferdinand de Saussure



Texto

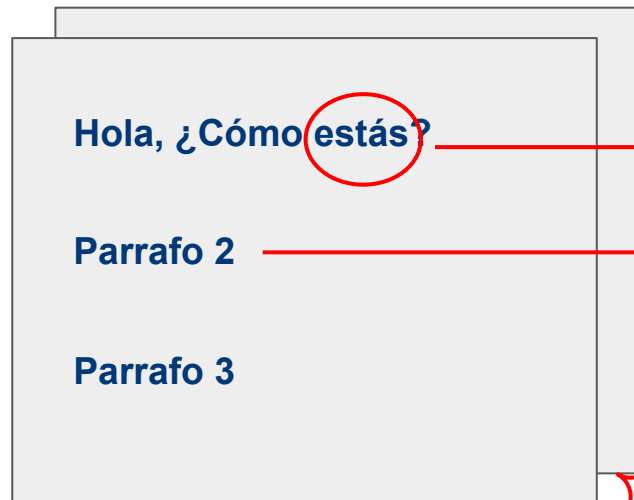
Timeline



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

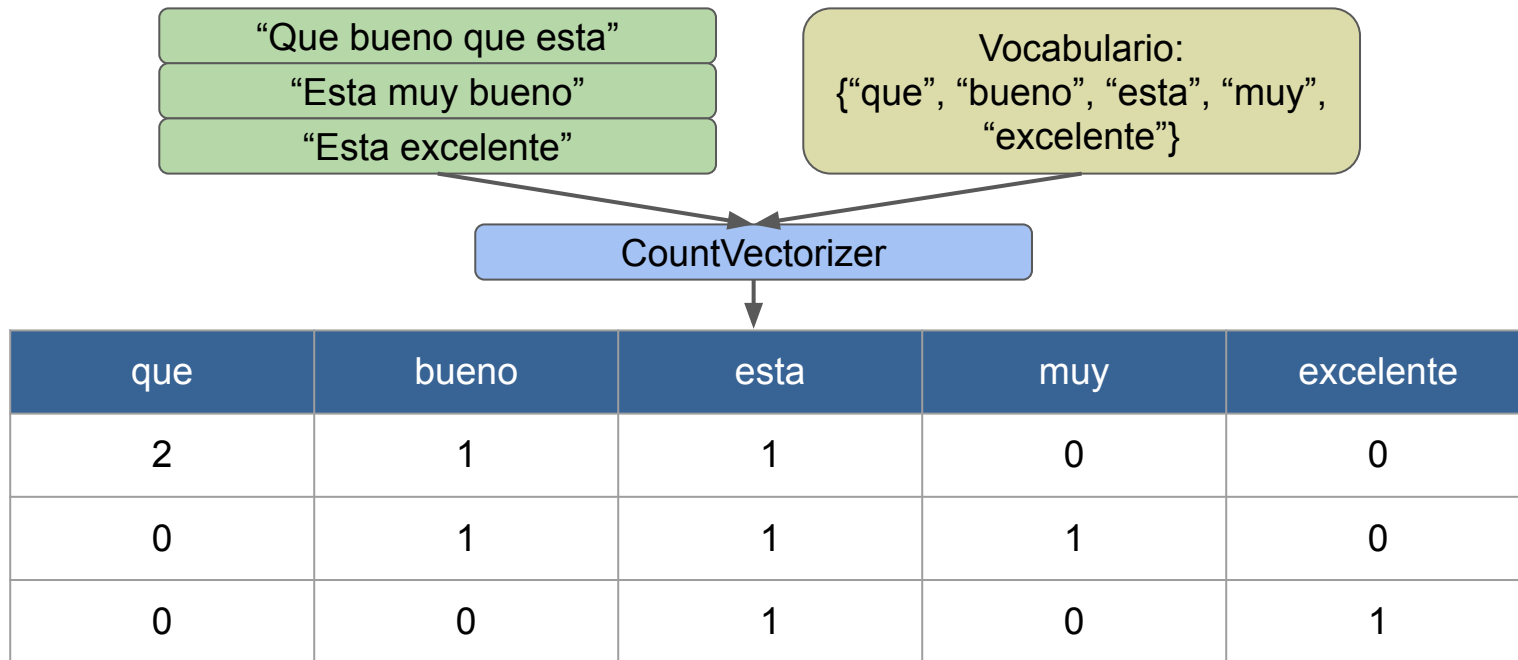


Vectores de
palabras/documentos

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"

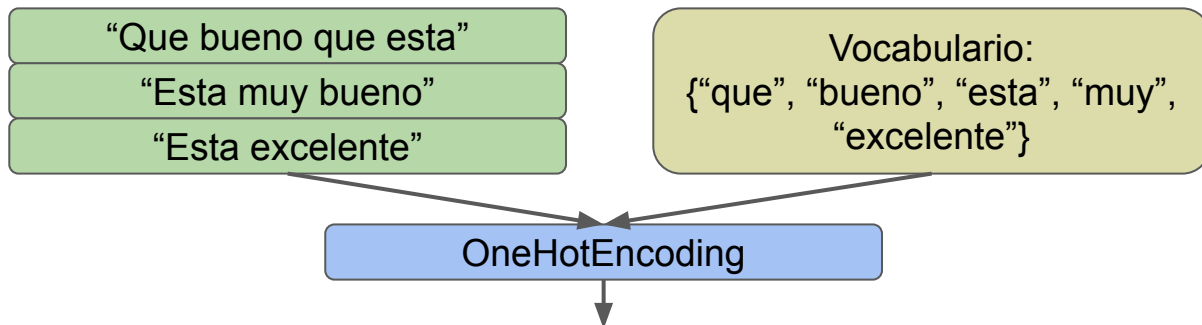


Los vectores tienen el tamaño del vocabulario

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 document frequency)



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

$$\text{TF-IDF}_{(n,d)} = \text{TF}_{(n,d)} \times \text{IDF}_{(n)}$$

Peso de un término (n) en un documento (d)

Frecuencia de aparición de un término (n) en un documento (d)

Factor IDF de un término (n)

Vector IDF (Inverse Document Frequency)



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

También suele utilizarse el logaritmo en base 2, su función es conseguir un coeficiente bajo, fácil de manejar

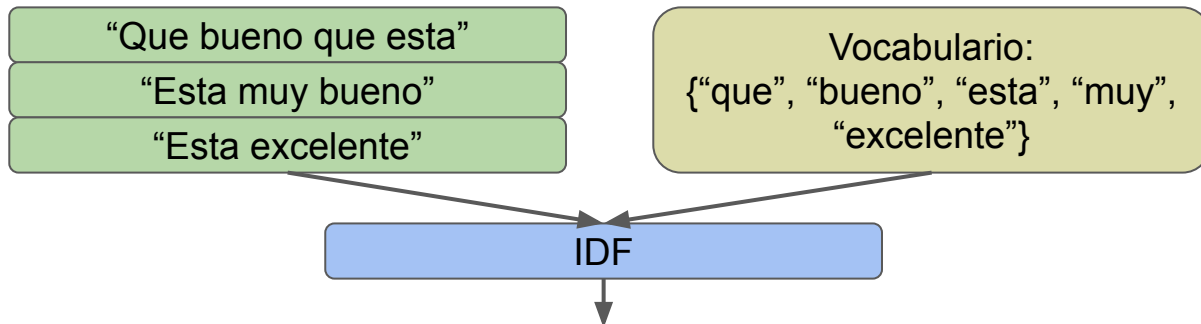
$$\text{IDF}_{(n)} = \log_{10} \frac{N}{\text{DF}_{(n)}}$$

N es el número total de documentos de la colección.

DF (Document Frequency) es el número documentos en los que aparece el término (n) a lo largo de toda la colección

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
$\log(3/1)$	$\log(3/2)$	$\log(3/3)$	$\log(3/1)$	$\log(3/1)$
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)



One-Hot Encoding

The quick brown fox jumped over the brown dog



	cat	the	quick	brown	fox	jumped	over	dog	bird	flew	...	kangaroo	house
time	0	1	0	0	0	0	0	0	0	0	...	0	0
	0	0	1	0	0	0	0	0	0	0	...	0	0
	0	0	0	1	0	0	0	0	0	0	...	0	0
	0	0	0	0	1	0	0	0	0	0	...	0	0
	0	0	0	0	0	1	0	0	0	0	...	0	0
	0	0	0	0	0	0	1	0	0	0	...	0	0
	0	1	0	0	0	0	0	0	0	0	...	0	0
	0	0	0	1	0	0	0	0	0	0	...	0	0
	0	0	0	0	0	0	0	1	0	0	...	0	0

Dictionary Size

¡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.

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 \rightarrow los vectores tienen la misma dirección.

Similitud coseno = 0 \rightarrow los vectores son ortogonales.

Similitud coseno = -1 \rightarrow los vectores apuntan en sentido contrario.

Intuición de la similitud coseno



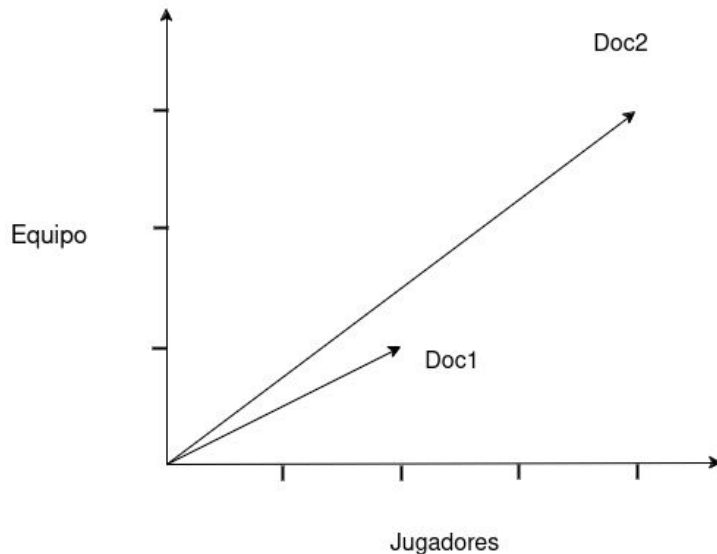
Doc1

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



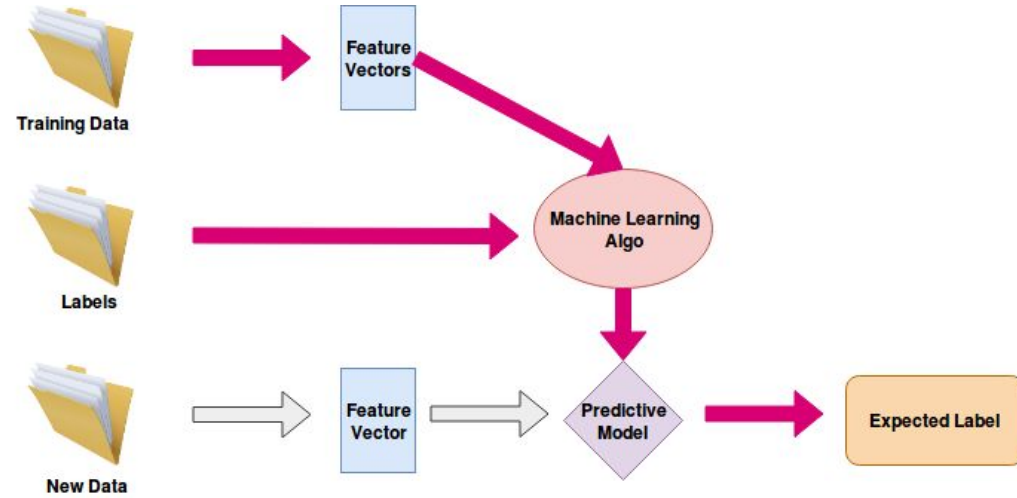
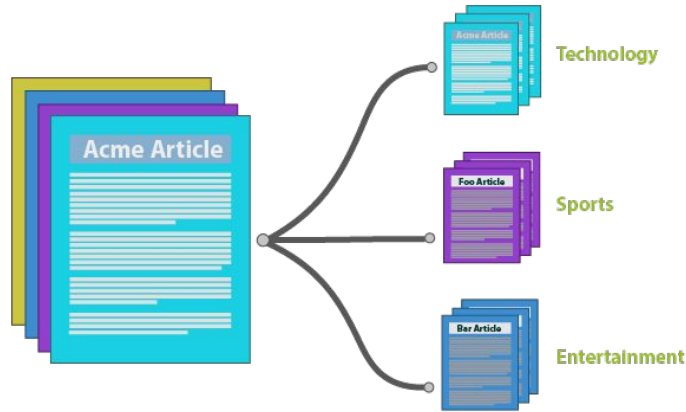
Doc2

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



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

Modelo de clasificación de texto





Modelo de clasificación Naïve Bayes

Se tiene un vocabulario $\{T_0, \dots, T_{V-1}\}$ de tamaño V y un corpus anotado de N documentos que se pueden clasificar en C clases. Cada documento d se representa como una sucesión $T_{j_1} T_{j_2} \dots T_{j_{n(d)}}$ o como un vector $[x_0, \dots, x_{V-1}]$.

Teorema de Bayes

$$\underbrace{P(C_i|d)} = \frac{P(d|C_i)P(C_i)}{P(d)}$$

Implementa un modelo probabilístico
de clasificación

Es un factor cte.

$$P(C_i|d) \propto \underbrace{P(d|C_i)P(C_i)}$$

Verosimilitud de los datos

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”

$$P(d|C_i) = P(T_{j_1}, \dots, T_{j_{n(d)}}|C_i) = \prod_{k=1}^{n(d)} P(T_{j_k}|C_i)$$

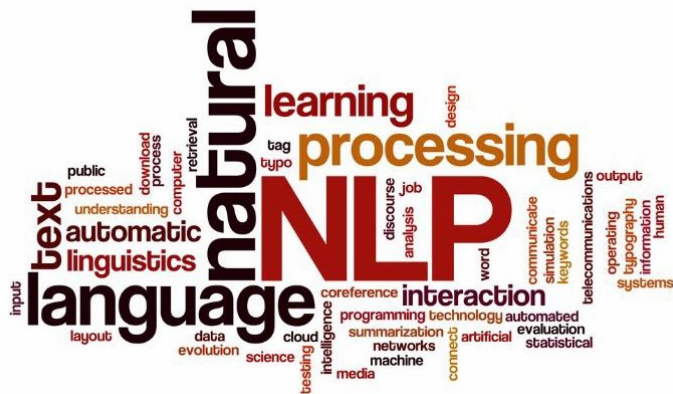
Sólo hay que calcular
la verosimilitud de
palabras por separado
dada la clase.

Modelo multinomial

$$P(d|C_i) = P([x_0, \dots, x_{V-1}]|C_i) = \frac{(x_0 + x_1 + \dots + x_{V-1})!}{x_0! \dots x_{V-1}!} \prod_{j=0}^{V-1} (P(T_j|C_i))^{x_j}$$

¡Bueno, bonito, barato!

[Explicación por sklearn](#)



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

Naïve Bayes es un ejemplo de clasificador tipo BOW



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.`