



UNIVERSIDAD
SAN SEBASTIAN

Herramientas básicas del PLN

Claudio Aracena, PhD
claudio.aracena@uss.cl

Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural (PLN) es un subcampo de la informática, especialmente de la inteligencia artificial. Su objetivo principal es dotar a las computadoras de la capacidad de procesar datos codificados en lenguaje natural.



Dimensionalidad

- Uno de los mayores desafíos del PLN es la dimensionalidad de su objeto de estudio.
- Para problemas con datos numéricos o categóricos, generalmente se tiene una cantidad limitada de dimensiones. Ej. edad, sexo, etc.
- Sin embargo, en datos textuales no es tan claro cuántas dimensiones podemos extraer. Puede ser la cantidad de palabras, oraciones, caracteres, etc.
- Dependiendo de la elección, es posible alcanzar miles de dimensiones.



El libro libro de Don Quijote la Mancha tiene:

Número total de palabras: 376.523

Número de palabras diferentes: 22.603

Bag of words

- Una de las formas más sencillas de representar un documento es mediante el modelo Bag of Words (BoW).
- Este modelo representa un documento como la conjunción de sus palabras y frecuencias, sin considerar orden.
- En este caso la dimensionalidad de un documento estará basada en el vocabulario del mismo (cantidad de palabras únicas).
- Si tenemos varios documentos (corpus) podemos representarlos como una matriz término-frecuencia

	1	2	3	4	5
clínica	3	0	2	0	0
cobertura	5	0	1	0	0
cobre	1	0	0	0	0
colesterol	1	0	0	0	0
conocimiento	1	0	0	0	0
contesta	1	0	0	0	0
contrato	2	0	1	0	5
correos	1	0	0	1	0
cumplir	2	0	0	0	0
doy	1	0	0	0	0
ejecutiva	2	0	1	0	0
eliminar	1	0	0	0	0
engañado	1	0	0	0	0
engaño	1	0	0	0	0
etc	1	0	0	0	0

Tokenización

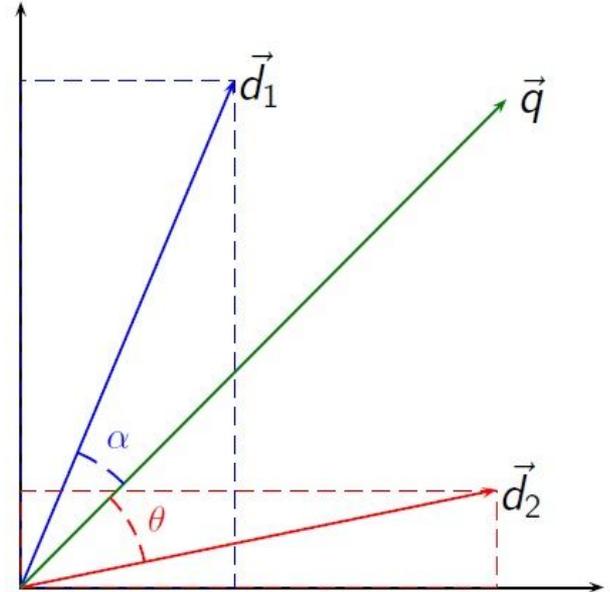
- Se refiere al proceso de separar una cadena de texto en tokens. Los tokens son unidades básicas de análisis textual. Estos pueden ser palabras o caracteres (de largo 1 o más).
- Ejemplos
 - la casa grande → ['la', 'casa', 'grande']
 - la casa grande → ['la', 'casa', 'gran', '##de']
 - la casa grande → ['l', 'a', ' ', 'c', 'a', 's', 'a', ' ', 'g', 'r', 'a', 'n', 'd', 'e']

Reducción de dimensionalidad

- Para reducir la dimensionalidad del problema, es posible aplicar las siguientes transformaciones:
- **Eliminar stopwords:** Se refiere al proceso de eliminar tokens de poco valor para el análisis que se realiza. En general, stopwords se refieren a preposiciones, artículos o verbos como hacer, haber, tener, etc.
- **Stemming:** Se refiere al proceso de llevar una palabra a su raíz no necesariamente morfológica. Ejemplos: niño → niñ; niña → niñ

Vector Space Model

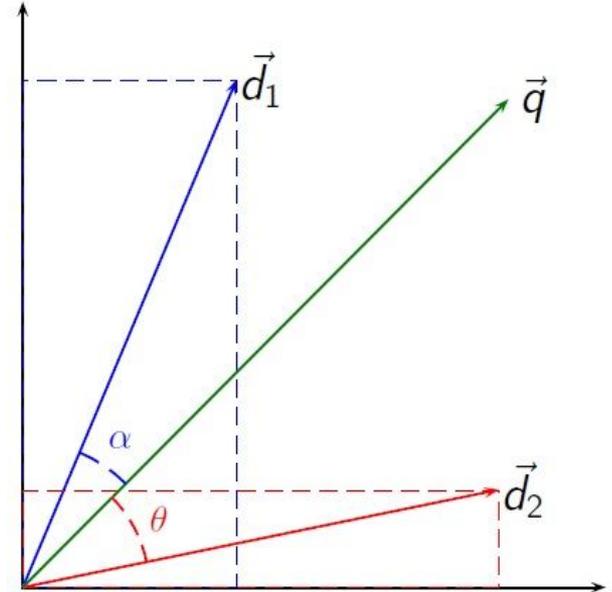
- A partir del modelo Bag of Words, podemos pensar los documentos como vectores que se encuentran en un espacio multidimensional.
- El número de dimensiones estará basado en el tamaño del vocabulario.
- Este modelo nos permite realizar comparación entre documentos



Vector Space Model

- Ejemplo
 - d1: el gato corre detrás del ratón
 - d2: el perro corre detrás del gato
 - q: ¿quién corre detrás del ratón?

palabra	documento 1	documento 2	query
perro	0	1	0
gato	1	1	0
ratón	1	0	1
corre	1	1	1



Term Frequency Inverse Document Frequency (TF-IDF)

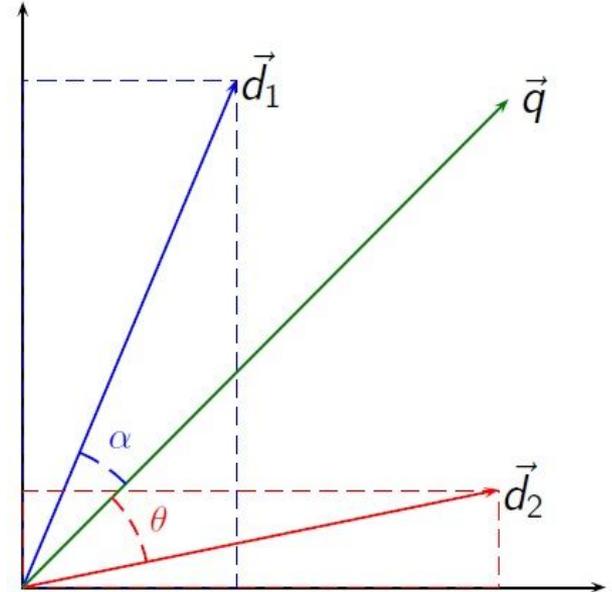
- Es una medida de la importancia relativa de un término dentro de un documento dado un corpus.
- Consiste en la multiplicación entre dos factores
 - El primer factor es la frecuencia del término dentro del documento (tf)
 - El segundo factor corresponde a la frecuencia inversa de los documentos que contienen al término dentro del corpus (idf)

$$tf \ idf = \frac{f_t}{\sum_{i=1}^T f_i} \times \log \frac{N}{df_t}$$

Vector Space Model con TF-IDF

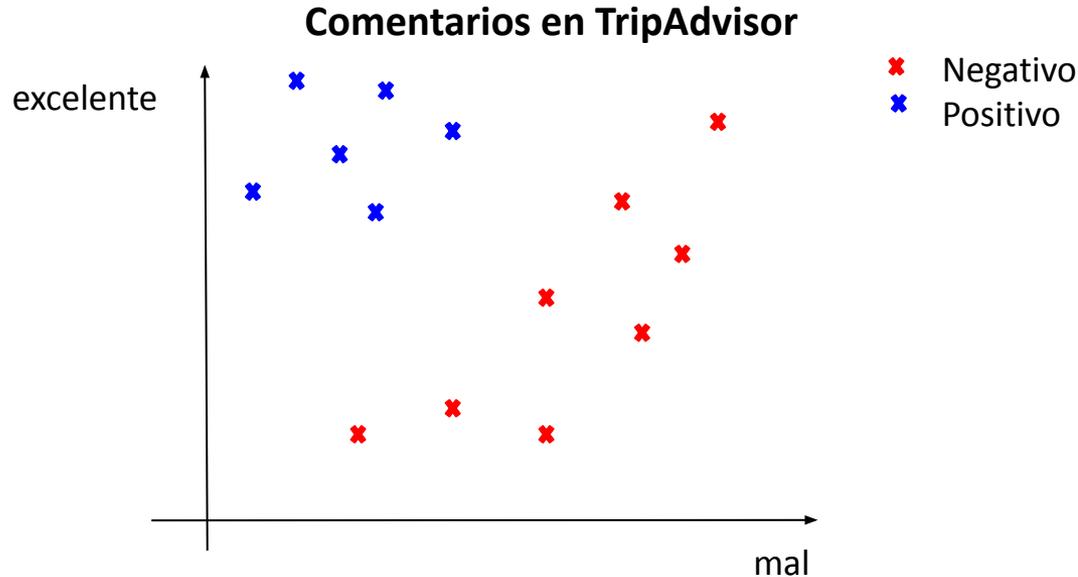
- Ejemplo
 - d1: el gato corre detrás del ratón
 - d2: el perro corre detrás del gato
 - q: ¿quién corre detrás del ratón?

palabra	TF-IDF doc 1	TF-IDF doc 2	TF-IDF query
perro	0	0.366	0
gato	0.135	0.135	0
ratón	0.135	0	0.202
corre	0	0	0



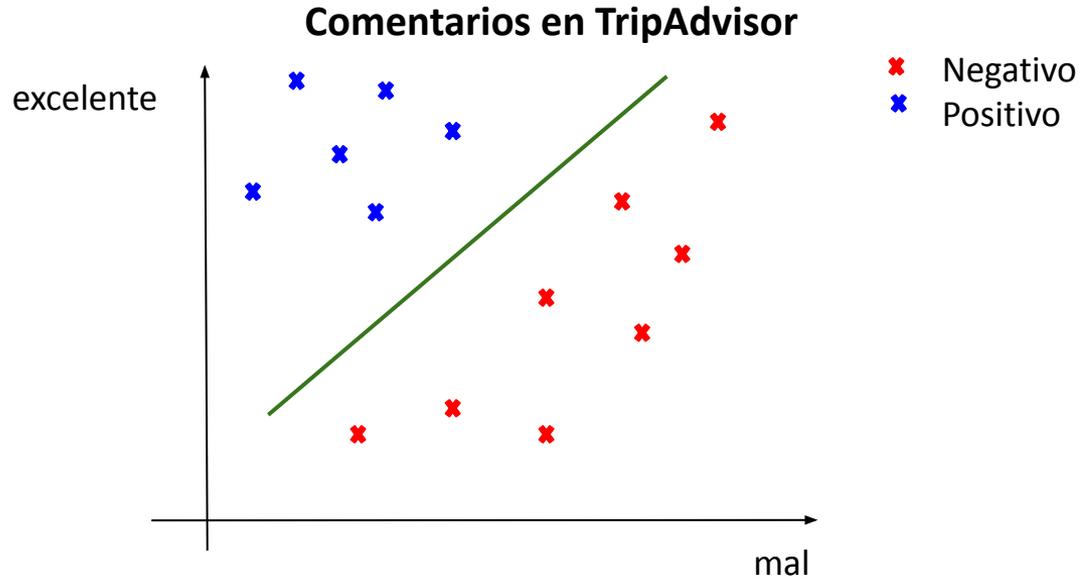
Clasificación de documentos

- Al igual que con datos numéricos o categóricos, es posible realizar clasificaciones en datos textuales utilizando modelos de Machine Learning.



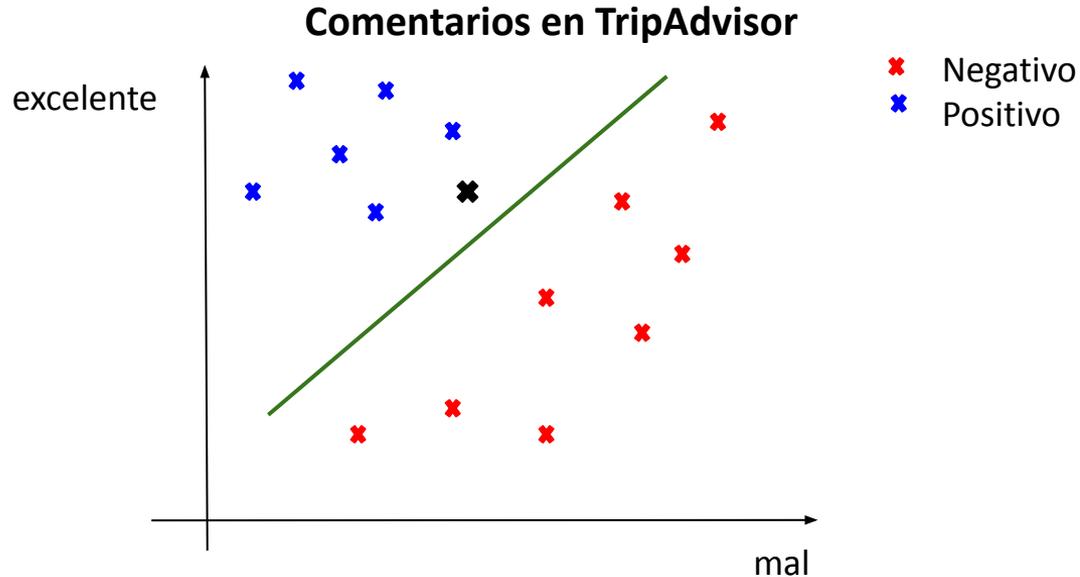
Clasificación de documentos

- Al igual que con datos numéricos o categóricos, es posible realizar clasificaciones en datos textuales utilizando modelos de Machine Learning.



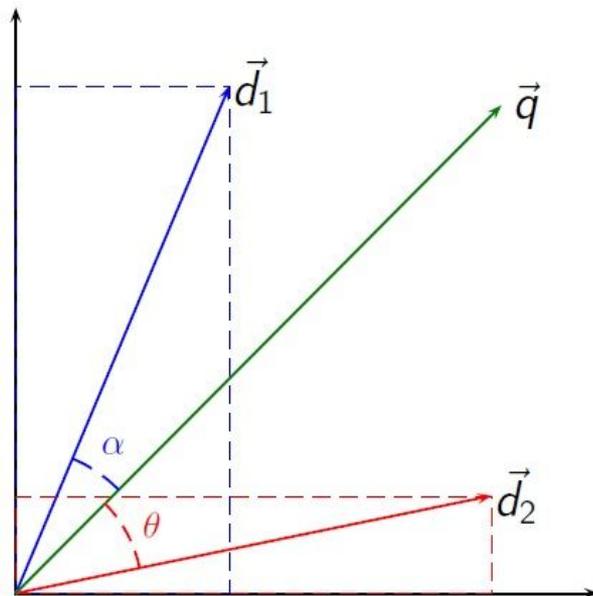
Clasificación de documentos

- Al igual que con datos numéricos o categóricos, es posible realizar clasificaciones en datos textuales utilizando modelos de Machine Learning.



Clasificación de documentos

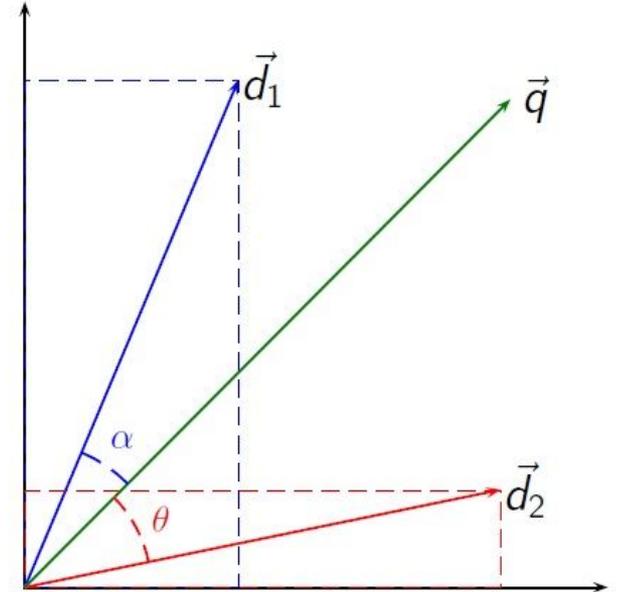
- Al igual que para casos anteriores, debemos hacer una representación vectorial de los documentos.
- Pero cada vector en el entrenamiento estará asociado a una clase.
- Las dimensiones del vector pueden ser la frecuencia de las palabras, tf-idf, u otras.



Clasificación de documentos

- Ejemplo
 - d1: un buen documento
 - d2: un mal documento

documentos	buen	mal	documento	clase
TF-IDF doc 1	0.347	0	0	buena
TF-IDF doc 2	0	0.347	0	mala

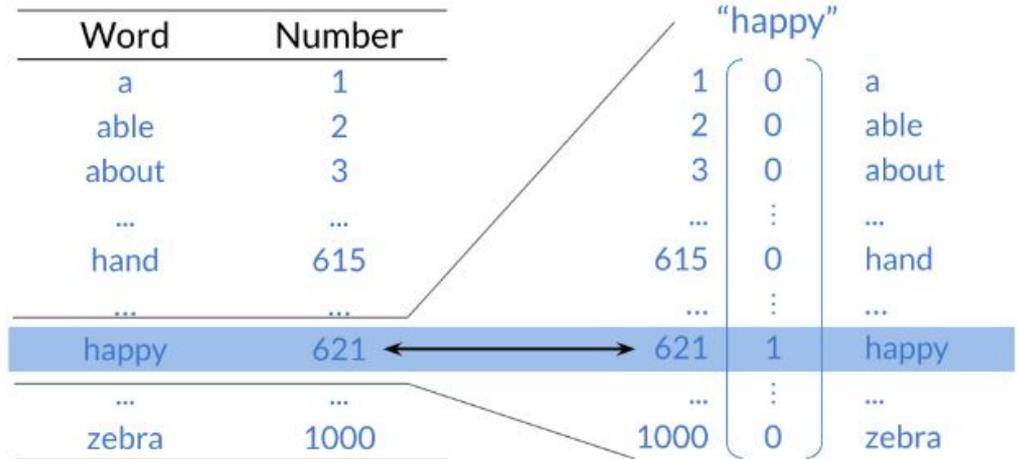


Words embeddings

Representación de palabras

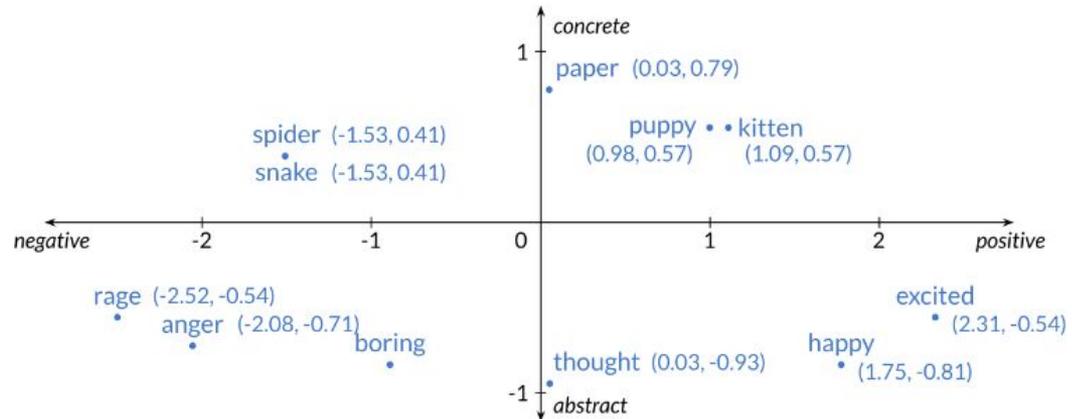
Las representaciones de palabras pueden ser clasificadas en:

- Números
- one-hot encoding
- word embeddings



Words embeddings

- Los word embeddings tienen la ventaja de tener una dimensionalidad menor a la del vocabulario y cada dimensión puede ser interpretada como una característica de la palabra

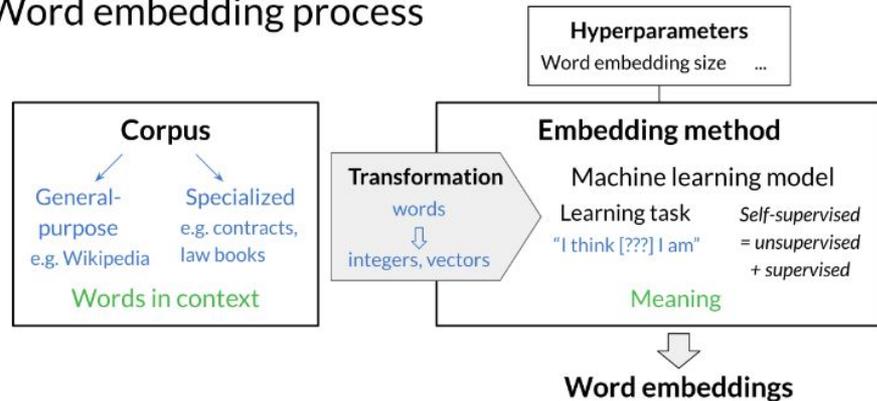


Extraído desde NLP with Probabilistic Models desde Coursera

Words embeddings

- Se necesitan 2 elementos para generar word embeddings:
 - corpus
 - método de embedding

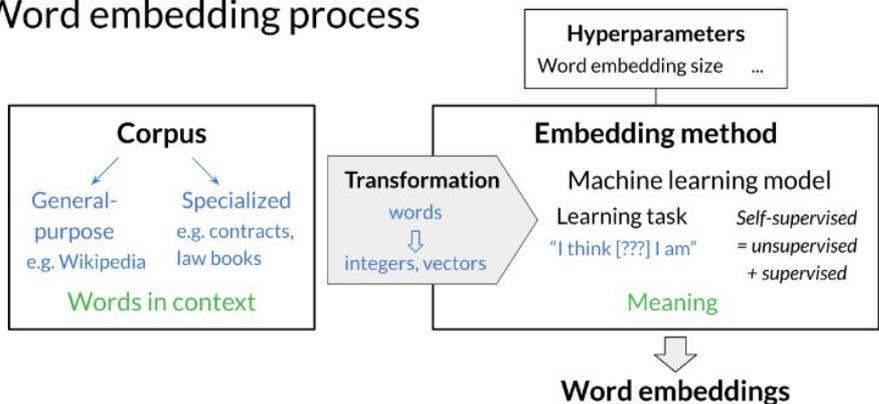
Word embedding process



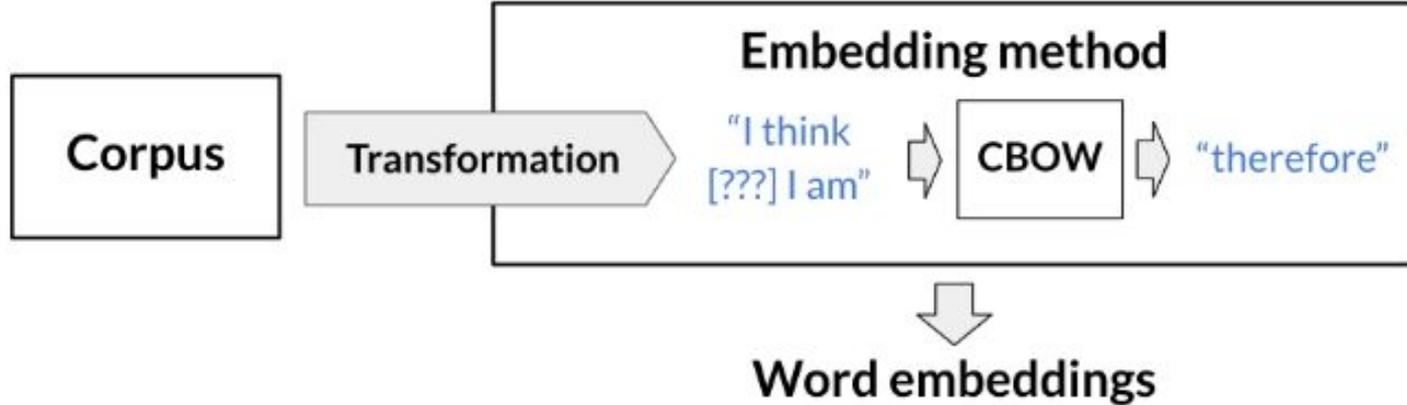
Words embeddings

- Se necesitan 2 elementos para generar word embeddings:
 - Corpus
 - Método de embedding
- Métodos de embedding:
 - CBOW (Google, 2013)
 - Skip-gram (Google, 2013)
 - Fast-text (Facebook, 2016)

Word embedding process



Continuous Bag of Words Model (CBOW)



Extraído desde NLP with Probabilistic Models desde Coursera

Words embeddings

- Para evaluar cuán bien resultaron los word embeddings se puede:
 - clusterizar
 - buscar vecinos más cercanos de palabras
 - medir distancias entre palabras sinónimas
 - medir distancias entre analogías: París es a Francia lo que Londres es a ...



Words embeddings

- Uno de los elementos para generar word embeddings es el corpus.
- Para español se han generado distintos corpus y se han entrenado word embeddings sobre ellos.
- Uno de los primeros corpus fue el [Spanish Billion Word Corpus](#).
- Otros corpus y embeddings pueden ser encontrados en:
<https://github.com/dccuchile/spanish-word-embeddings>.

Modelos de lenguaje

Modelos de Lenguaje

Un modelo de lenguaje asigna una probabilidad a una secuencia de palabras.

- Por ejemplo, la frase “En un lugar de la Mancha” es mucho más probable que la frase “lugar En la un Mancha de”.

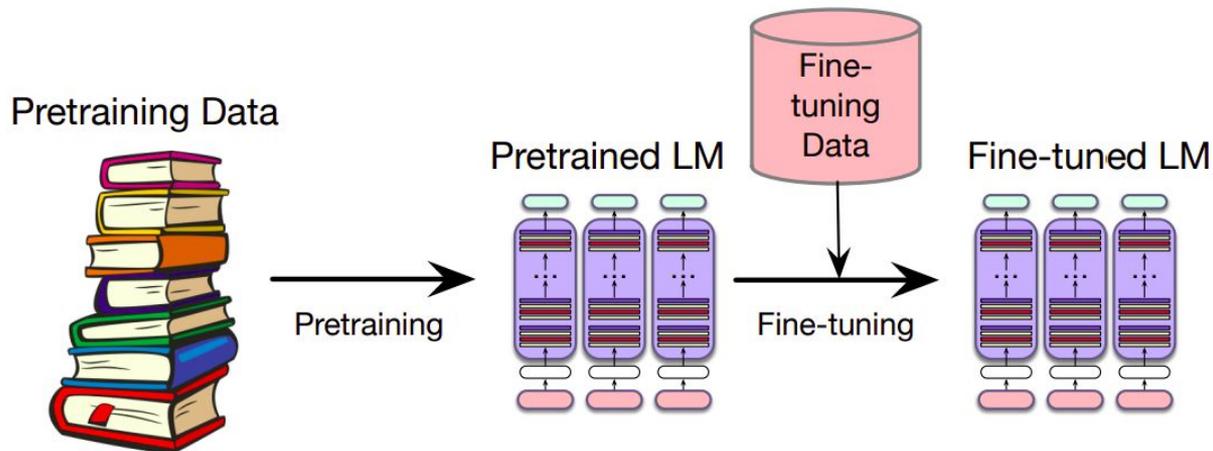
También un modelo de lenguaje puede asignar una probabilidad de aparición a la siguiente palabra en una frase.

- Por ejemplo, en la frase “el cielo es”, un modelo de lenguaje asignaría mayor probabilidad a la palabras “azul” o “celeste” que a las palabras “motocicleta” o “roncar”.

Los modelos de lenguaje son útiles para generar texto, traducir, corregir errores y detección del habla.

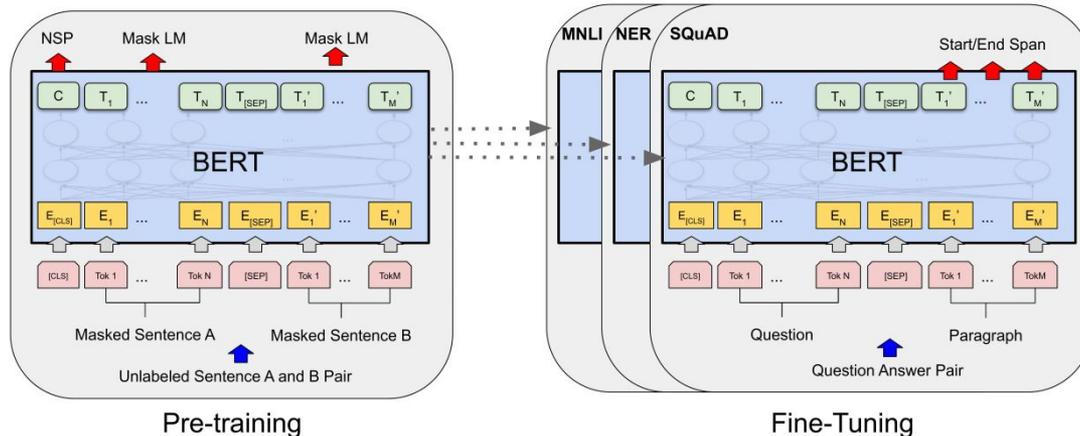
Modelos de Lenguaje pre-entrenados

- Un modelo de lenguaje pre-entrenado utiliza un corpus para tener una base de entrenamiento y luego puede ser ajustado para una tarea específica como clasificación o reconocimiento de entidades.



BERT

- Bidirectional Encoder Representations from Transformers (BERT, 2018) fue uno de los primeros modelos de lenguaje pre-entrenados basados en la arquitectura de transformers.

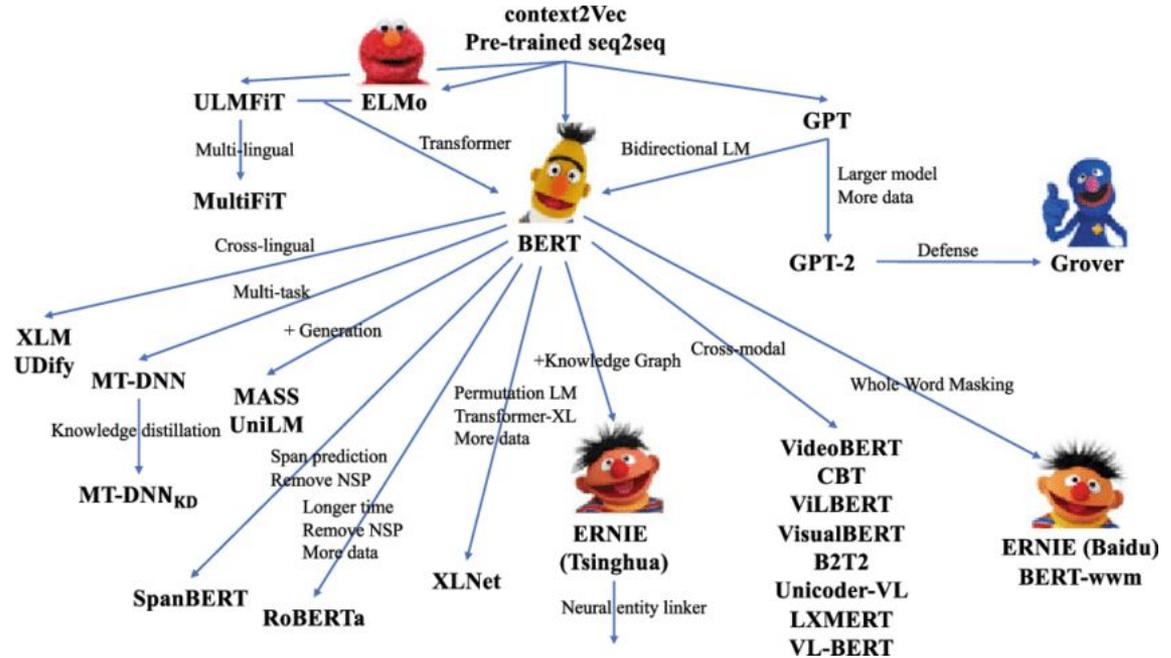


Extraído desde Devlin, J. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Modelos de Lenguaje pre-entrenados

- GPT
- RoBERTa
- DeBERTa

- En español
 - BETO
 - BERTIN
 - RoBERTa-bne



HuggingFace

- Repositorio de modelos de lenguaje pre-entrenados y datasets.
- Se puede probar los modelos habilitados.
- Se puede filtrar por tarea y lenguaje.
- Se puede compartir modelos de lenguaje pre-entrenados propios.
- Tiene varios recursos de aprendizaje
- <https://huggingface.co/>



HUGGING FACE