Introducción al Procesamiento del Lenguaje Natural (NLP)

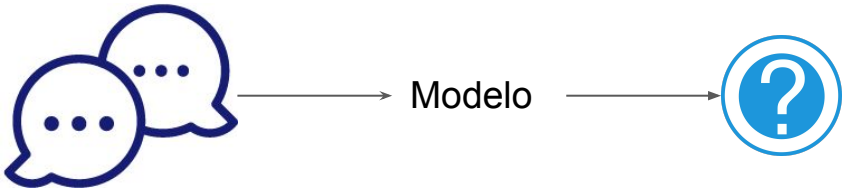
Pablo Lucero

https://www.linkedin.com/in/pablo-lucero-ec/

¿Que es el NLP?

Linguistica + Machine Learning

Clasificación de texto



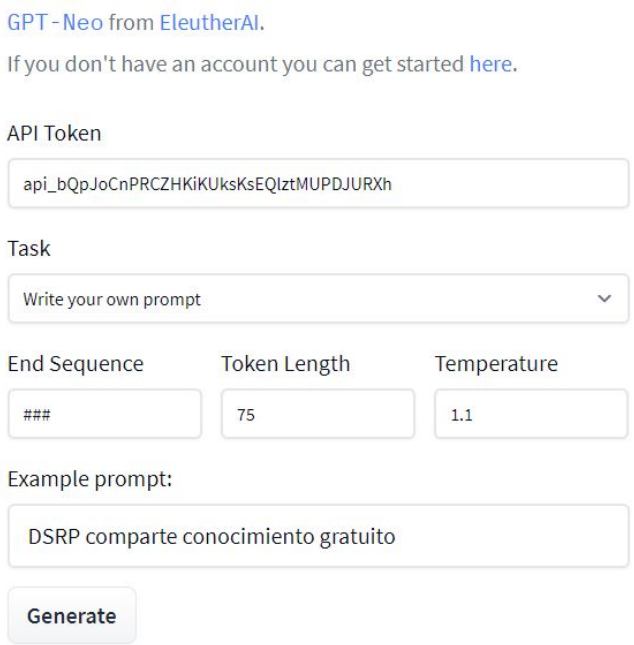
Clasificación de sentimientos

Clasificación de spam

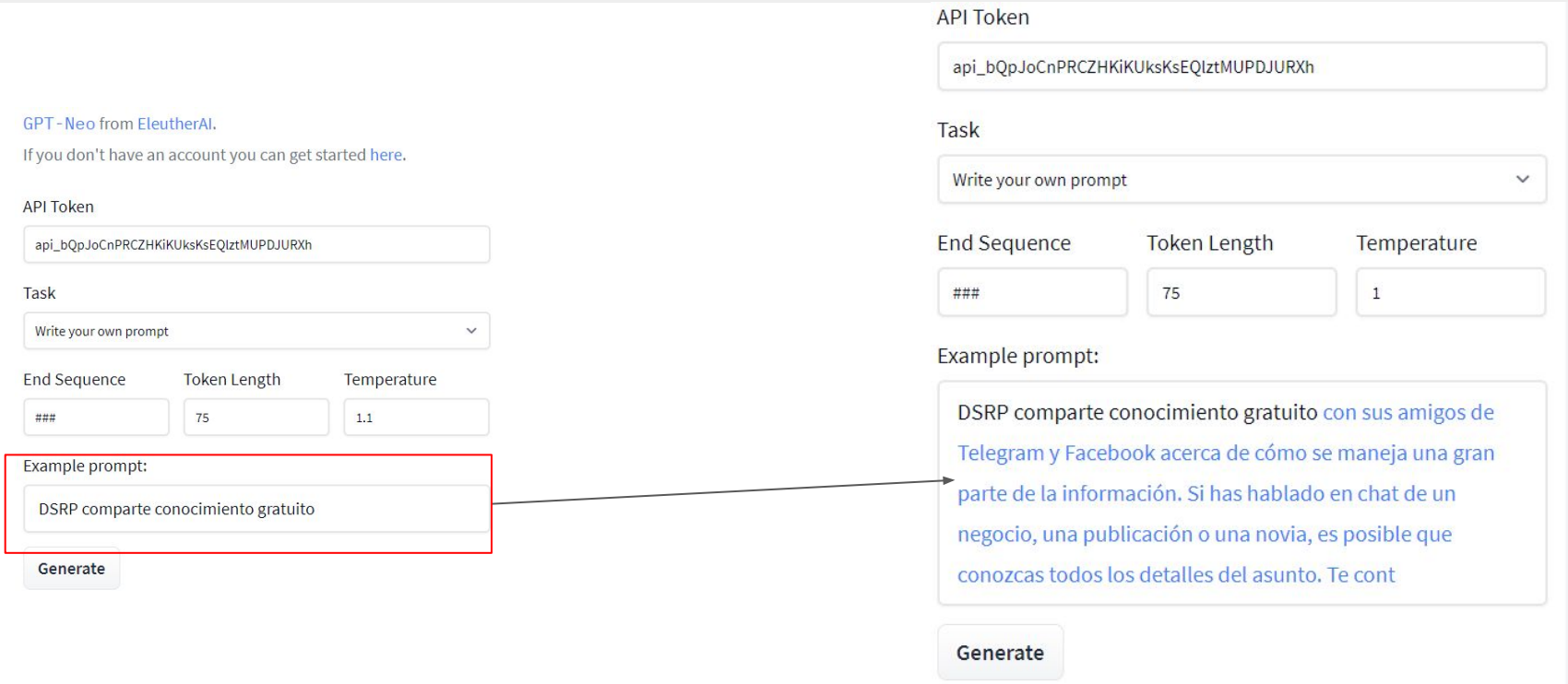
Generación de texto

GPT-3

GPT-Neo



Generación de texto



Entre otros dominios

Audio a texto

Texto a imagen
DALLE-OpenAI

Word embeddings

Word embeddings

Son **representaciones numéricas** del texto que pueden introducirse en un modelo de ML.

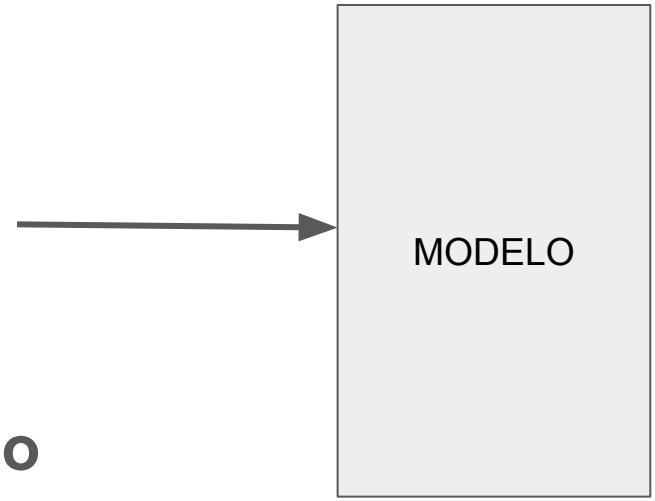
Word embeddings

Hay

un

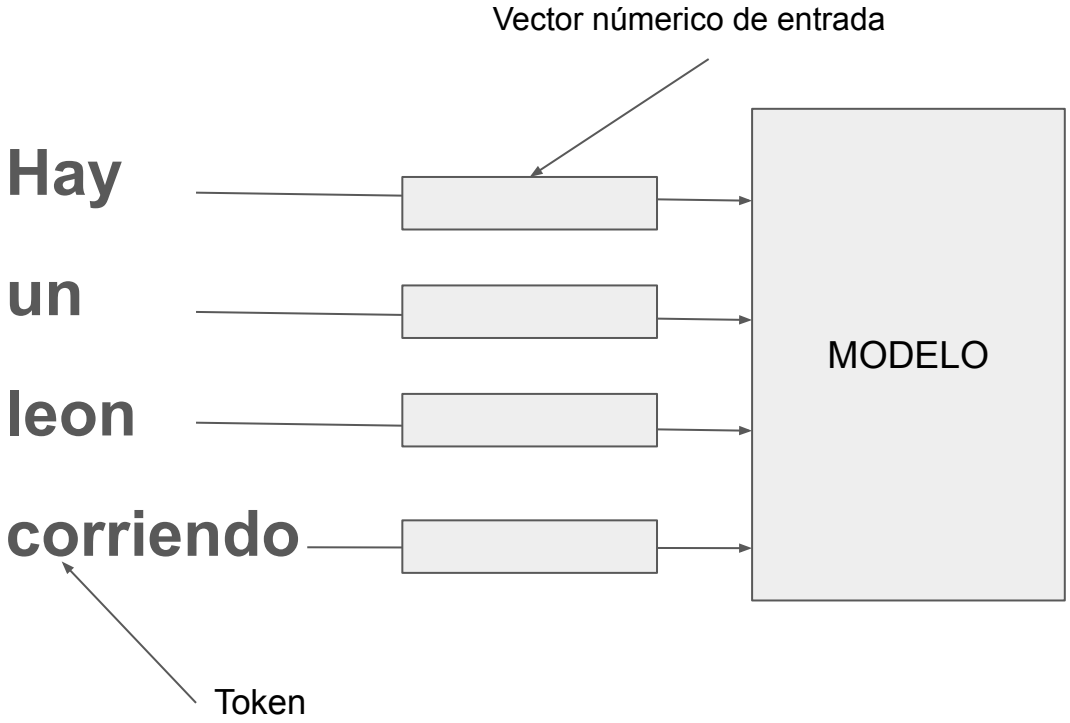
leon

corriendo

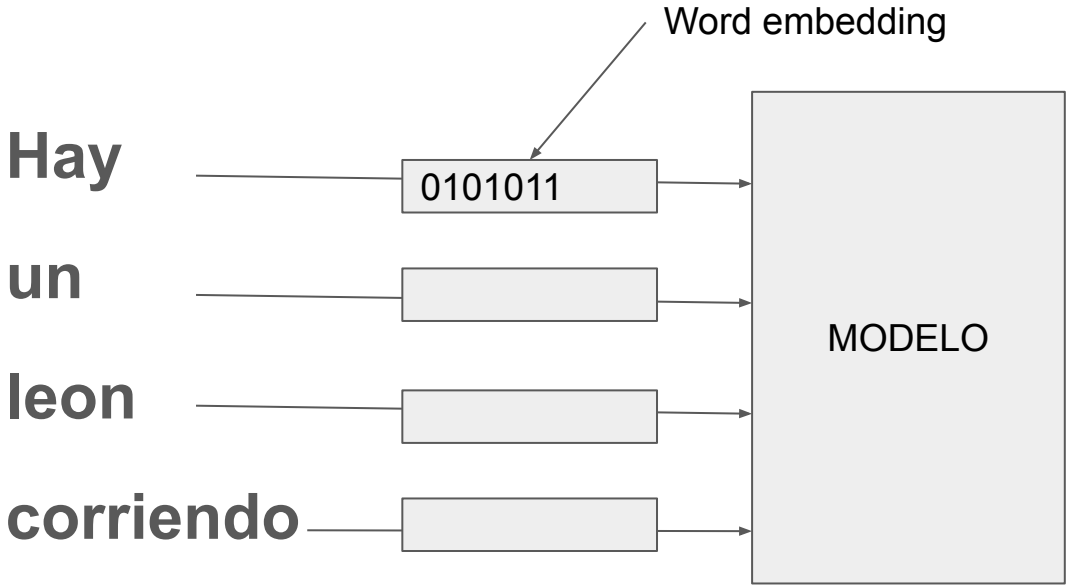


Un modelo de ML necesitará una representación numérica para poder ser entrenado.

Word embeddings (Tokenización)



Word embeddings



Word embeddings

¿Como sabemos que embedding corresponde a cada palabra?

Word embeddings

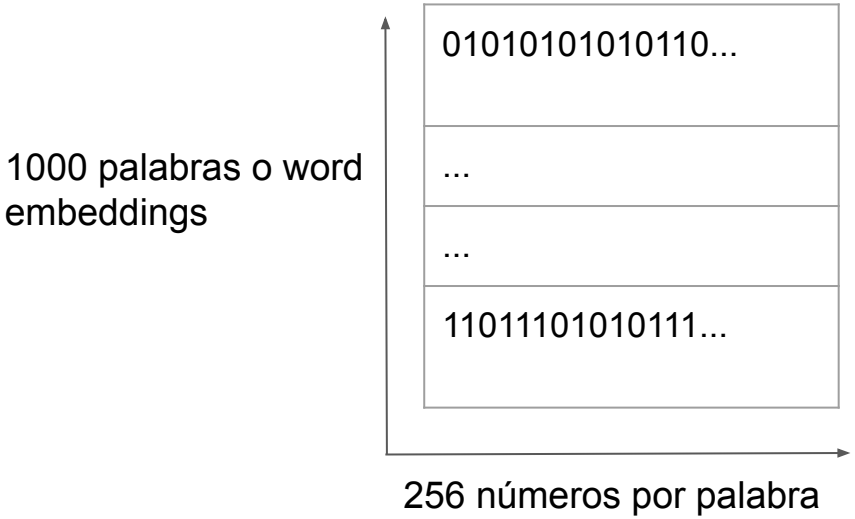
Se realiza una tabla de mapeo

idx	Palabra	Embedding
1	Un	01010101010110...
...	....	
42	Hay	11111101010100...
100	Leon	11011101010111...

Hay → **Idx 42** → Embedding 42

Word embeddings

¿Cómo representamos una oración de 1000 palabras?



Lematización y Radicalización (Stemming)

Lematización y Radicalización (Stemming)

Lemmatization: Permite obtener la forma básica de una palabra (sin conjugar y en singular) retirando y reemplazando los sufijos.

Ejemplo:

Corrí, Corremos, Correré = Correr
Bailé, Bailamos, bailaré= Bailar
Un, unos, unas = uno
Niño, Niña, Niñito, niñote = Niño

Nos permite normalizar nuestros tokens.

Lematización y Radicalización (Stemming)

Stemming: Permite reducir una palabra a su raíz (stem, en ingles), que es la forma invariante de palabras relacionadas.

Ejemplo:

Corrí, Corremos, Correré = Corr
Bailé, Bailamos, bailaré= Bail
Un, unos, unas = un
Niño, Niña, Niñito, niñote = Niñ

Nos permite normalizar nuestros tokens.

Bag of words

Bag of words (Bolsa de palabras)

Es una representación del texto que permite describir la aparición de las palabras dentro de un documento.

Se inicia construyendo un vocabulario de palabras únicas y una medida de la presencia de estas palabras en un texto.

Bag of words (Bolsa de palabras)

Ejemplo:

- Review 1: This movie is very scary and long.
- Review 2: This movie is not scary and is slow.
- Review 3: This movie is spooky and good.

Vocabulario:

1. This
2. movie
3. is
4. very
5. scary
6. and
7. long
8. not
9. slow
10. spooky
11. good

Bag of words (Bolsa de palabras)

Ejemplo:

- Review 1: This movie is very scary and long.
- Review 2: This movie is not scary and is slow.
- Review 3: This movie is spooky and good.

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Bag of words (Bolsa de palabras)

Inconvenientes

1. Si las nuevas oraciones contienen nuevas palabras, entonces el tamaño de nuestro vocabulario aumentaría y, por lo tanto, la longitud de los vectores también aumentaría.
2. Además, los vectores también contendrían muchos ceros, lo que resultaría en una matriz escasa (qué es lo que nos gustaría evitar)
3. No retenemos información sobre la gramática de las oraciones ni sobre el orden de las palabras en el texto.

TF-IDF (Frecuencia de término-Frecuencia inversa de documentos)

TF-IDF (Frecuencia de término-Frecuencia inversa de documentos)

Es una representación del texto que permite describir la aparición de las palabras dentro de un documento.

Frecuencia de término (TF): Es una medida de la frecuencia con la que aparece un término o palabra, t, en un documento, d.

Frecuencia inversa de documentos (IDF): Es una medida de la importancia de un término. Esto significa qué tan común o rara es una palabra en todo el conjunto de documentos.

TF-IDF (Frecuencia de término-Frecuencia inversa de documentos)

Ahora podemos calcular la puntuación TF-IDF para cada palabra del corpus. Las palabras con una puntuación más alta son más importantes y las que tienen una puntuación más baja son menos importantes:

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t$$

TF-IDF (Frecuencia de término-Frecuencia inversa de documentos)

Ejemplo:

- Review 1: This movie is very scary and long.
- Review 2: This movie is not scary and is slow.
- Review 3: This movie is spooky and good.

Term	Review 1	Review 2	Review 3	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.000	0.000	0.000
movie	1	1	1	0.000	0.000	0.000
is	1	2	1	0.000	0.000	0.000
very	1	0	0	0.068	0.000	0.000
scary	1	1	0	0.025	0.022	0.000
and	1	1	1	0.000	0.000	0.000
long	1	0	0	0.068	0.000	0.000
not	0	1	0	0.000	0.060	0.000
slow	0	1	0	0.000	0.060	0.000
spooky	0	0	1	0.000	0.000	0.080
good	0	0	1	0.000	0.000	0.080

Word to Vec

Word to Vec

Word2Vec es un método para construir word embeddings (representaciones vectoriales de palabras).

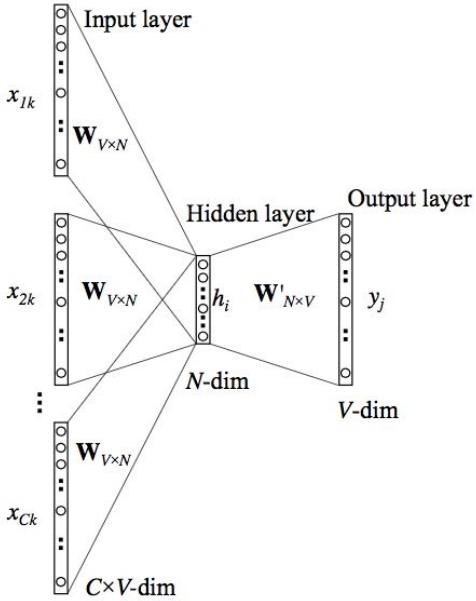
Estas representaciones se pueden obtener utilizando dos métodos (ambos con redes neuronales): Skip Gram y Continuos Bag Of Words (CBOW).

Word to Vec

Continuous Bag of Words: Predice la palabra objetivo (verde) sumando los vectores de contexto.

Word to Vec

Continuous Bag of Words: Modelo



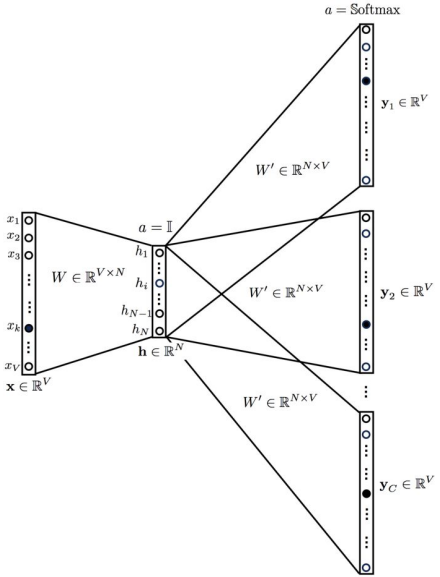
De esta forma se generan representaciones de palabras usando las palabras de contexto

Word to Vec

Skip Gram: Predice el contexto (rojo) dado una palabra central (verde).

Word to Vec

Skip Gram: Modelo



De esta forma se generan representaciones de palabras usando las palabras centrales.

¡GRACIAS!

Bibliografia

[1] Introduction to Natural Language Processing (NLP)

[2] Natural Language Processing (NLP) using Python

[3] Word embeddings

[4] Lematizacón y Stemming

[5] Bag of Words

[6] Word2Vec

[7] word2vec Parameter Learning Explained

[8] Curso de NLP de cero a cien