Introducción al Procesamiento de Lenguaje Natural

Grupo de PLN – InCo

Redes Neuronales

y

vectores de palabras

La importancia de la semántica

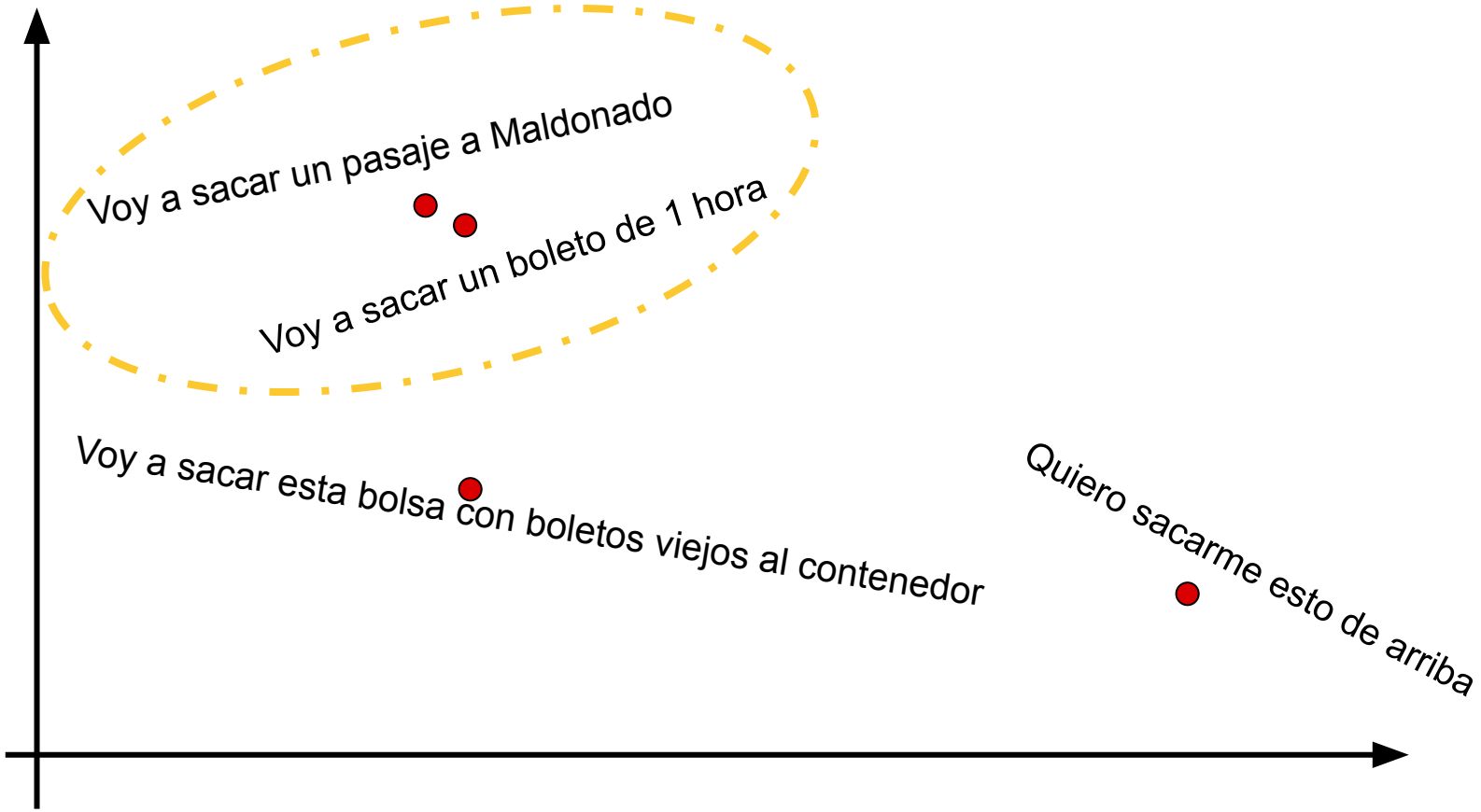
(en PLN)

1. Voy a sacar un pasaje a Maldonado
2. Quiero sacarme esto de arriba
3. Voy a sacar esta bolsa con boletos viejos al contenedor
4. Voy a sacar un boleto de 1 hora

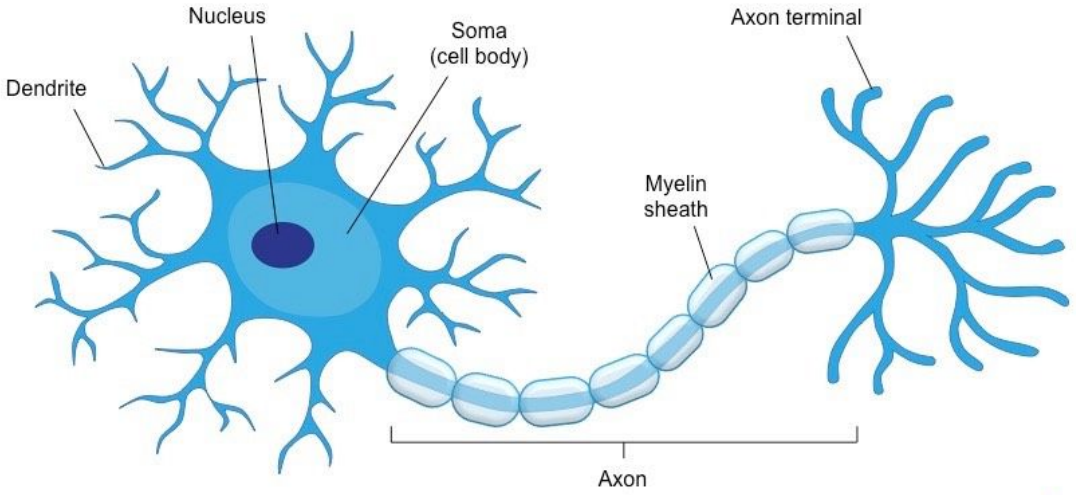
Problema de clasificación: determinar si la entrada del usuario es válida para un asistente de compra de viajes

La importancia de la semántica

(en PLN)

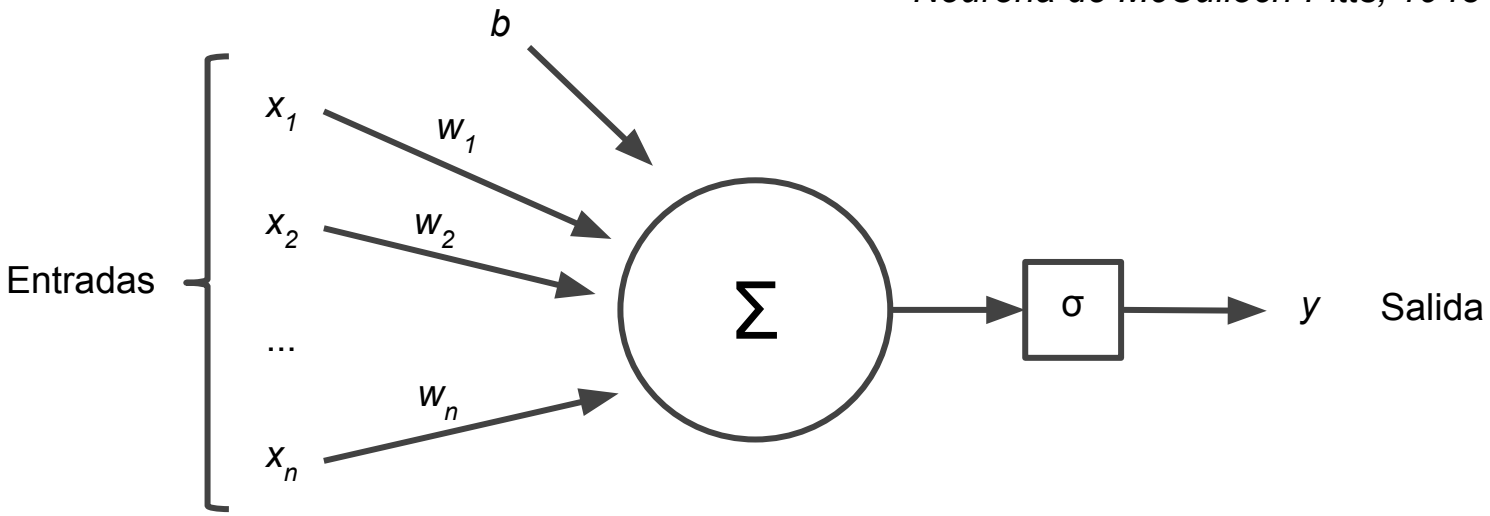


Redes Neuronales



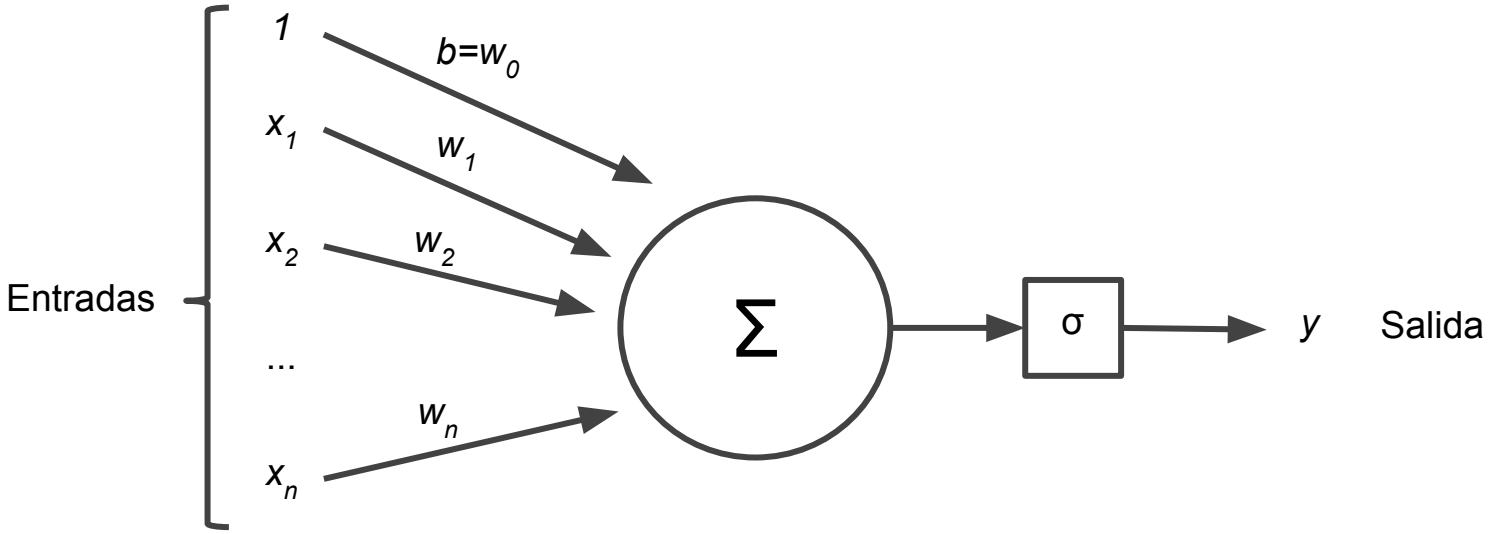
Redes Neuronales

Neurona de McCulloch-Pitts, 1943



$$y = \sigma(\sum_i x_i w_i + b)$$

Redes Neuronales



$$\hat{x} = [1, x_1, x_2, ..., x_n]$$
$$\hat{w} = [w_0, w_1, w_2, ..., w_n]$$

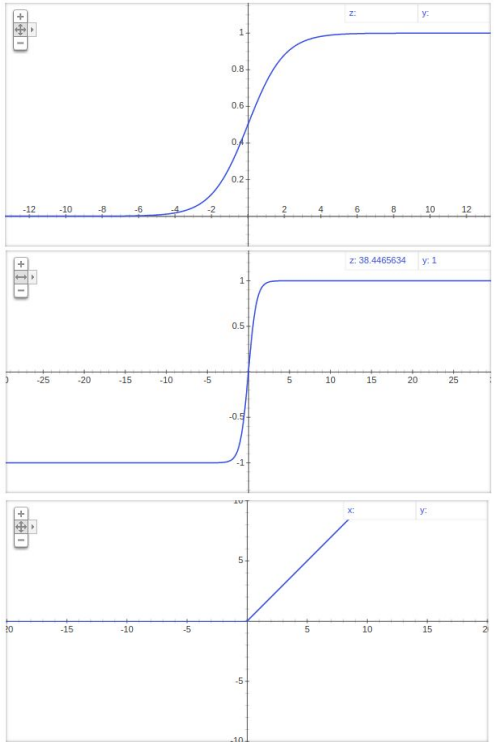
$$y = \sigma(\hat{x}.\hat{w})$$

Redes Neuronales

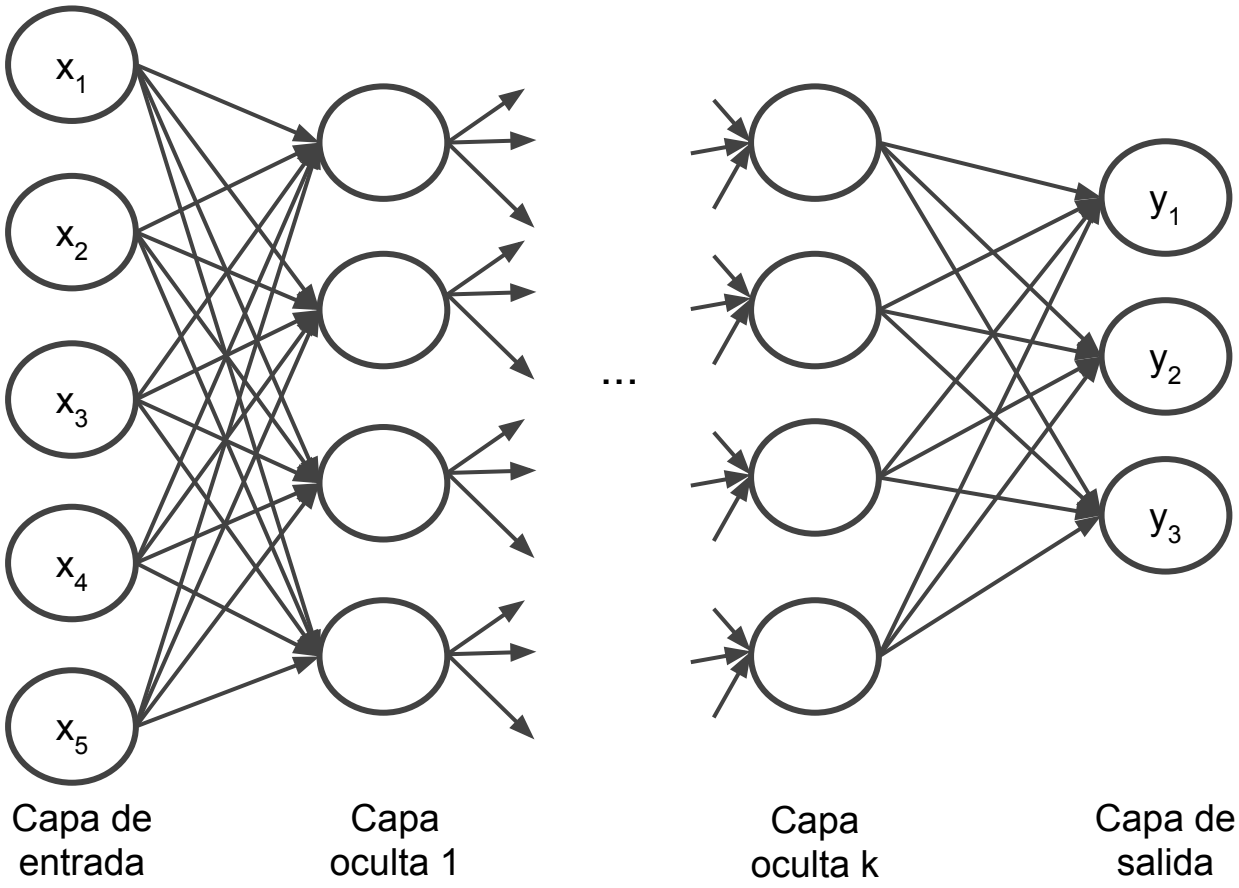
Funciones de activación

En lo posible: derivables, crecientes y no lineales

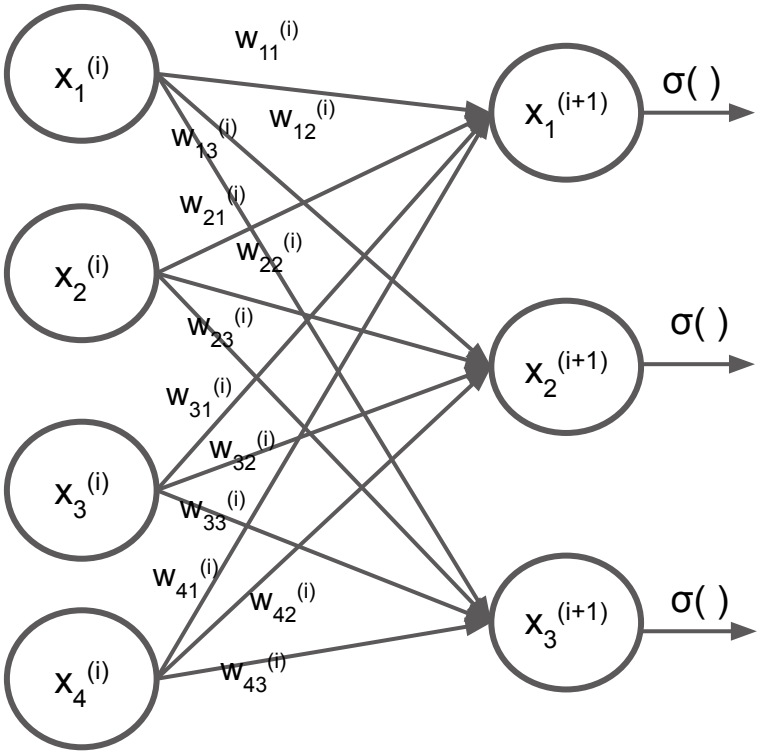
- Función sigmoide o logística: $\sigma(z) = \dfrac{1}{1+e^{-z}}$
- Tangente hiperbólica: $\tanh(z) = \dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
- ReLU: $\mathrm{relu}(z) = max(0, z)$
- Otras...



Redes Neuronales



Redes Neuronales



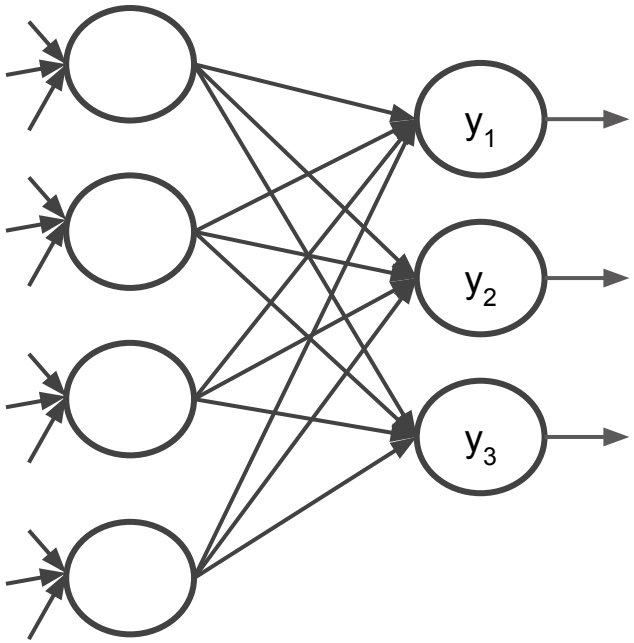
Entrada: $x^{(i)} = [\, x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)} \,]$

Salida: $x^{(i+1)} = [\, x_1^{(i+1)}, x_2^{(i+1)}, x_3^{(i+1)} \,]$

$$W^{(i)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix}$$

$$x^{(i+1)} = \sigma\left(x^{(i)} W^{(i)}\right)$$

Redes Neuronales



Última capa oculta

Capa softmax

Para problemas de clasificación en clases discretas, queremos que la salida sea una distribución de probabilidad.

Para eso se suele utilizar una capa softmax

$$P(j|x) = \frac{e^{y_j}}{\sum_k e^{y_k}}$$

j es una de las k clases

Entrenamiento

Funciones de pérdida

¿Qué tanta discrepancia hay entre los valores esperados y los valores predichos por la red?

Se requiere: derivables, y que sean expresivas midiendo cuánto y_i discrepa con respecto a $\hat{y}_i$

N instancias x_i de dimensión M

Valores esperados y_i

Valores predichos $\hat{y}_i$ (resultado de la red)

- Error Cuadrático Medio:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

- Entropía Cruzada:

$$CE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(\hat{y}_{ij})$$

→ y si es para valores categóricos:

$$CE = -\frac{1}{N}\sum_{i=1,y_j=1}^{N} log(\hat{y}_{ij})$$

Entrenamiento

Se trata de encontrar los pesos que minimicen la función de pérdida

- Descenso estocástico por gradiente
  - puro (de a un ejemplo)
  - con *minibatches* (varios ejemplos)
- Backpropagation (cómo calcular el gradiente eficientemente)

¿Puedo encontrar la mejor función?

- Óptimos locales
- Sobreajuste

Vectores de palabras

- En PLN trabajamos principalmente con **texto**
- Las RN y la mayoría de los clasificadores utilizan valores numéricos como entrada, por lo que necesitamos una **representación numérica** de textos
  - palabras
  - oraciones
  - documentos
- Es deseable que esta representación numérica tenga propiedades explotables (e.g. un resultado de *distancia* interpretable)

Hipótesis distribucional

En los 1950s surge la hipótesis distribucional (Firth):

Palabras que aparecen en contextos similares tienden a tener significados similares

La milanesa con queso más rica es la uruguaya.

Sí, es re rica la hamburguesa con queso de ese lugar.

A la milanesa con queso mozzarella y salsa le decimos napolitana.

El otoño es una de las estaciones del año.

¡El verano es una de mis estaciones favoritas!

En invierno hace pila de frío.

En verano nunca hace frío.

Matriz término-Término

Representa las palabras contando las palabras que las rodean, según un **contexto**.

El **contexto** puede ser el documento entero (archivo, tweet, página web o lo que sea)
pero lo más común es tomar **N palabras de ventana**.

O sea, si *X* es la palabra a modelar: $\text{palabra}_{-N} \ldots \text{palabra}_{-2}\text{palabra}_{-1}\ X\ \text{palabra}_{1}\text{palabra}_{2} \ldots \text{palabra}_{N}$

¿Cómo quedaría la matriz con el ejemplo anterior y usando N=4?

La milanesa con queso más rica es la uruguaya.

Sí, es re rica la hamburguesa con queso de ese lugar.

A la milanesa con queso mozzarella y salsa le decimos napolitana.

El otoño es una de las estaciones del año.

¡El verano es una de mis estaciones favoritas!

En invierno hace pila de frío.

En verano nunca hace frío.

	...	rica	queso	frío	estaciones	...
...						
milanesa		1	2	0	0	
hamburguesa		1	1	0	0	
otoño		0	0	0	0	
verano		0	0	1	0	
invierno		0	0	1	0	
...						

Matriz término-Término

Representa las palabras contando las palabras que las rodean, según un **contexto**.

El **contexto** puede ser el documento entero (archivo, tweet, página web o lo que sea)
pero lo más común es tomar **N palabras de ventana**.

O sea, si *X* es la palabra a modelar: $palabra_{-N} \ldots palabra_{-2} palabra_{-1}\ X\ palabra_{1} palabra_{2} \ldots palabra_{N}$

¿Cómo quedaría la matriz con el ejemplo anterior y usando N=5?

La milanesa con queso más rica es la uruguaya.

Sí, es re rica la hamburguesa con queso de ese lugar.

A la milanesa con queso mozzarella y salsa le decimos napolitana.

El otoño es una de las estaciones del año.

¡El verano es una de mis estaciones favoritas!

En invierno hace pila de frío.

En verano nunca hace frío.

	...	rica	queso	frío	estaciones	...
...						
milanesa		1	2	0	0	
hamburguesa		1	1	0	0	
otoño		0	0	0	1	
verano		0	0	1	1	
invierno		0	0	1	0	
...						

PROBLEMA → **los vectores son enormes y con muchos ceros (*sparse*)**

Word2Vec

En 2013 Mikolov et al. propusieron **word2vec**, un par de algoritmos para crear colecciones de vectores de palabras **densos** (con pocos 0s) y de baja dimensionalidad (típicamente entre 150 o 300).

Idea: en vez de contar las palabras en una ventana de contexto, entrenemos un clasificador que prediga qué tan probable es que la palabra **c** aparezca en el contexto de **w**. O sea: $P(+|w,c)$

Como queremos que las palabras más relacionadas tengan vectores cercanos y los de las menos relacionadas estén más lejos necesitamos **ejemplos negativos**.

Técnica de ***negative sampling***: elegir palabras que no compartan contexto con **w**. Por cada ejemplo positivo (**w**,$\mathbf{c}_{pos}$) tomamos *k* ejemplos negativos (**w**,$\mathbf{c}_{neg}$).

Word2Vec: ejemplo de *sampling*

Un **juego de rol**, conocido en español como **JDR** (de las siglas en inglés RPG, que significa "role-playing game" o "juego de interpretación de roles"), es un tipo de juego en el cual los participantes adoptan e interpretan diferentes roles o personajes. En estos juegos, cada jugador asume el papel de un personaje jugador (abreviado comúnmente como "PJ"), lo que implica adoptar su identidad, motivaciones y objetivos.

Un grupo de jugadores de rol participando en una sesión privada en un domicilio particular.

Secuencias de tokens para **ventana = 2**

["un", "juego", "**de**", "rol", "conocido"]
["juego", "de", "**rol**", "conocido", "en"]
["de", "rol", "**conocido**", "en", "español"]
["rol", "conocido", "**en**", "español", "como"]
["conocido", "en", "**español**", "como", "JDR"]

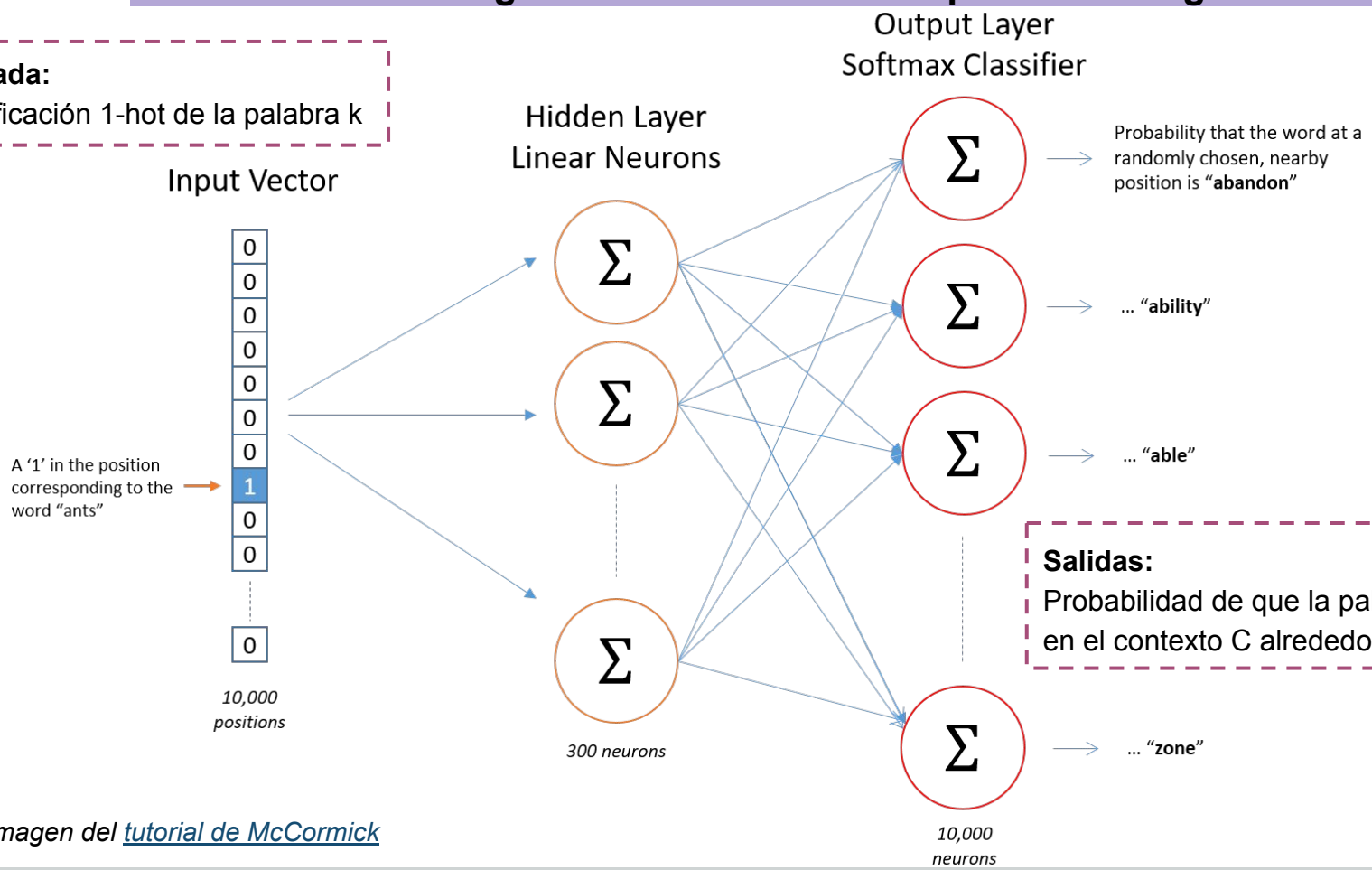
Negative sampling con ***k=2***

("**de**", "rol")	("**de**", "raíz") ("**de**", "muy")
("**de**", "conocido")	("**de**", "mar") ("**de**", "cebolla")
("**rol**", "conocido")	("**rol**", "hoy") ("**rol**", "bien")
("**rol**", "en")	("**rol**", "ya") ("**rol**", "mesa")

Word2Vec: Algoritmo skip-gram

skip-gram intenta modelar las palabras más probables que aparecerán alrededor de una palabra. **Los word embeddings son el estado de la capa oculta luego del entrenamiento**

Entrada:
Codificación 1-hot de la palabra k



Salidas:
Probabilidad de que la palabra j esté en el contexto C alrededor de la palabra k

Imagen del [tutorial de McCormick]

Word2Vec

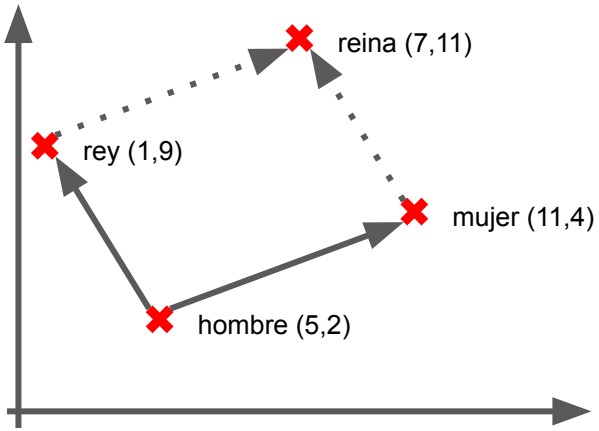
Se asocia una palabra (string) a un vector de reales

Vectores más cercanos tienden a ser semánticamente similares (similaridad coseno)

"Descubre" relaciones entre palabras

rey – hombre + mujer ≅ reina

uruguay – montevideo + francia ≅ parís

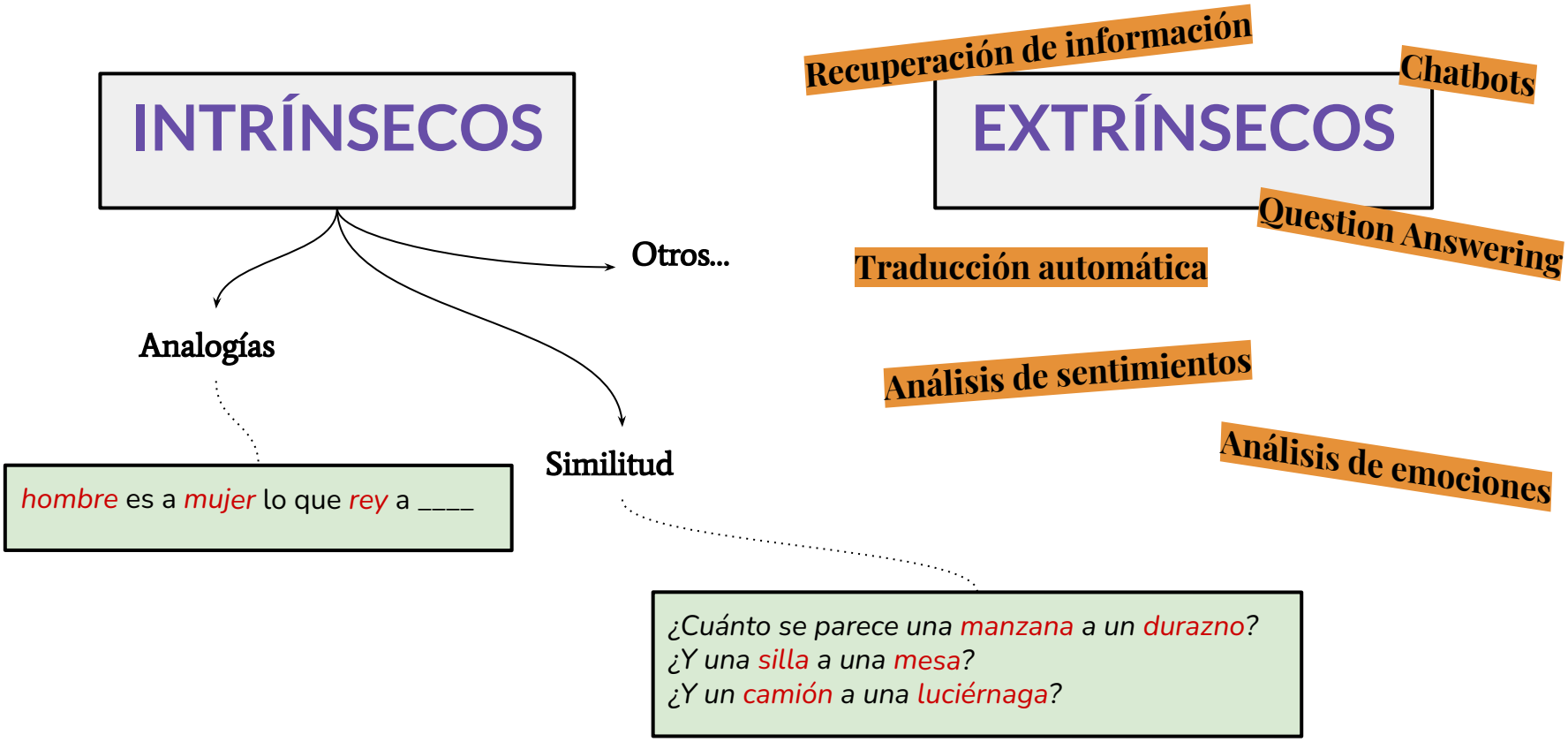


Se considera palabra a nivel de *string*, por lo que "vela" 🕯 y "vela" ⛵ van a estar representadas por el mismo vector

PROBLEMA → no hay distinción entre diferentes significados de una palabra

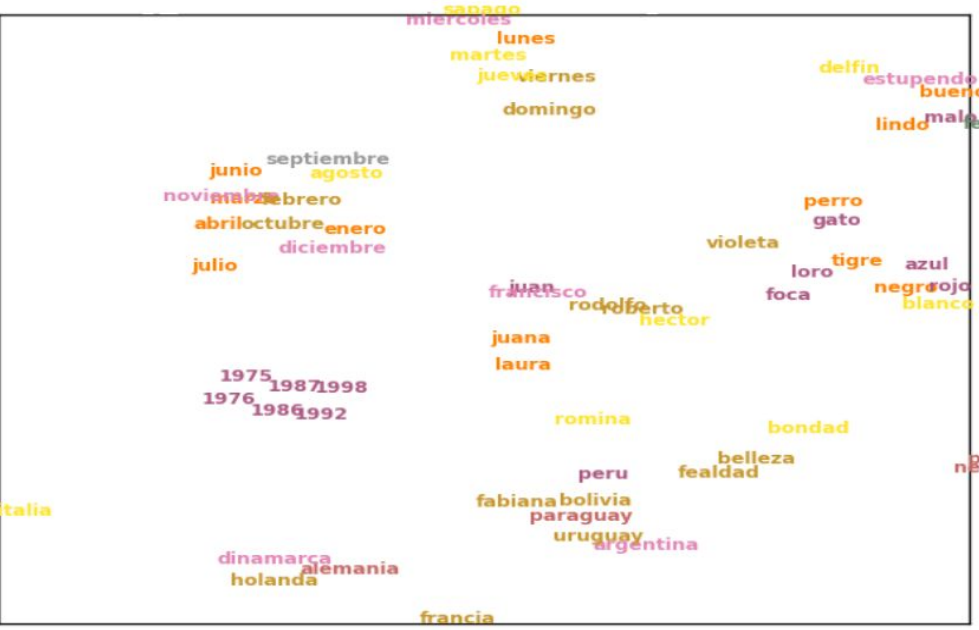
Evaluación

¿Cómo sabemos si una colección de embeddings está bien?

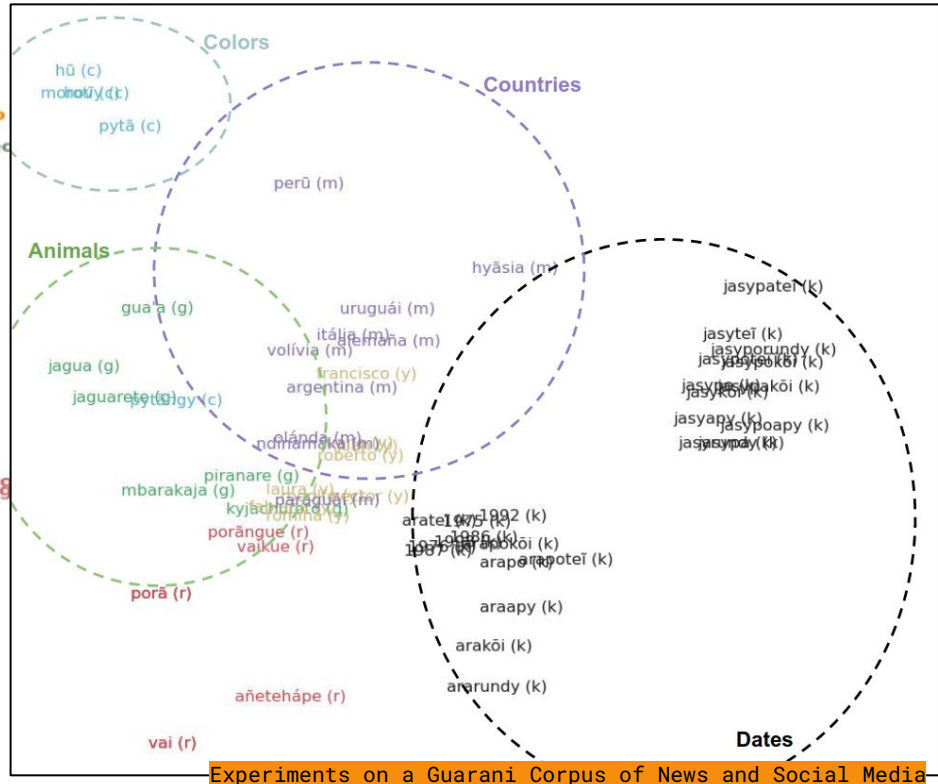


Evaluación

Visualización: reducir la dimensionalidad del espacio vectorial y graficar algunas palabras para ver cómo se agrupan



Spanish Word Vectors from Wikipedia (Etcheverry & Wonsever, LREC 2016)



Experiments on a Guarani Corpus of News and Social Media (Góngora, Giossa & Chiruzzo, AmericasNLP 2021)

Aprendizaje Profundo

Los *word embeddings* cambiaron la historia del PLN.

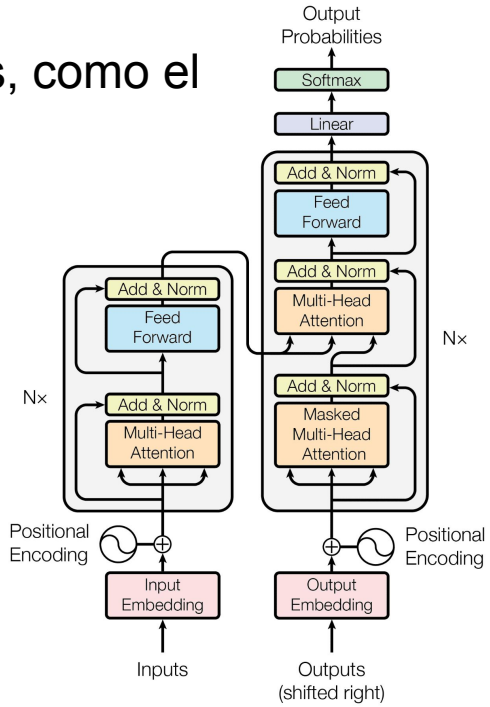
Con esta representación numérica de las palabras podemos…

Usar arquitecturas de redes neuronales más complejas, como el

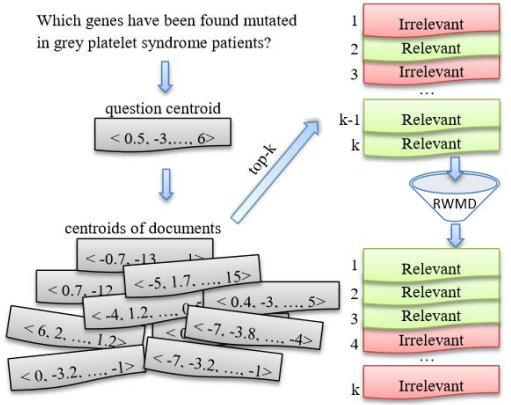
Transformer

BERT

GPT



Usarlos como *features* en métodos de aprendizaje clásicos, como el **centroide**



Using Centroids of Word Embeddings and Word Mover's Distance for Biomedical Document Retrieval in Question Answering (Brokos et al., BioNLP 2016)