

Redes Neuronales para el **análisis** y la **generación** de texto

Grupo PLN
InCo- Fing- Udelar



ELI - IV Escuela Latinoamericana de Informática
Octubre 2025 - Valparaíso

Word embeddings

Vectores de palabras

- En PLN trabajamos principalmente con **texto**.
- Las RRNN 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 (medir distancias).

Vectores de palabras

- Usual: atributos de tipo Bag of Words (BoW):
 - el vector es del tamaño del vocabulario
 - con 0 y 1
 - o cantidad de ocurrencias
 - o cantidad ponderada (tf/idf)
 - puedo eliminar stop words
 - puedo usar lemas o raíces (stemming)
- Vector de atributos que representan características del texto:
cantidad de palabras positivas/negativas, largo del texto,
cantidad de adjetivos, ...
- Word embeddings

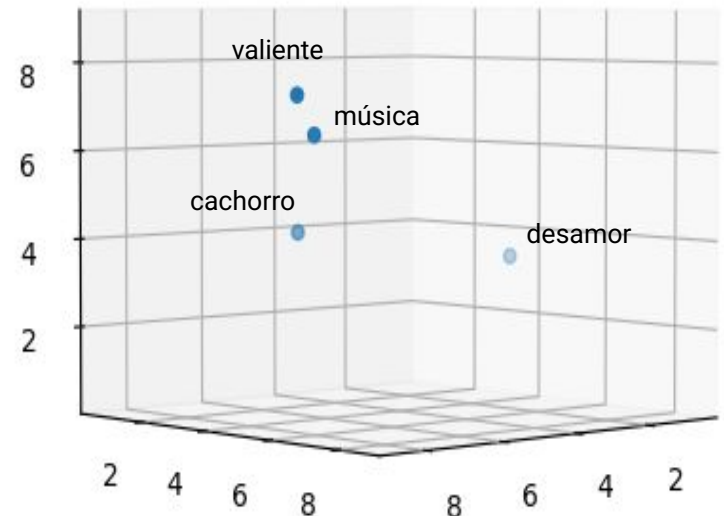
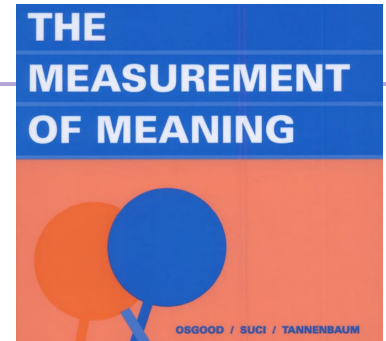
Vectores de palabras

Antecedente histórico: Osgood et al. (1957) proponen que el contenido afectivo de las palabras se descompone en tres dimensiones:

- *valencia*
- *entusiasmo*
- *dominancia*

Con estos valores, cada palabra podría representarse como un vector de tres dimensiones

	<i>valencia</i>	<i>entusiasmo</i>	<i>dominancia</i>
valiente	8,05	5,50	7,38
música	7,67	5,57	6,50
desamor	2,45	5,65	3,58
cachorro	6,71	3,95	4,24



Vectores de palabras

Antecedente histórico:

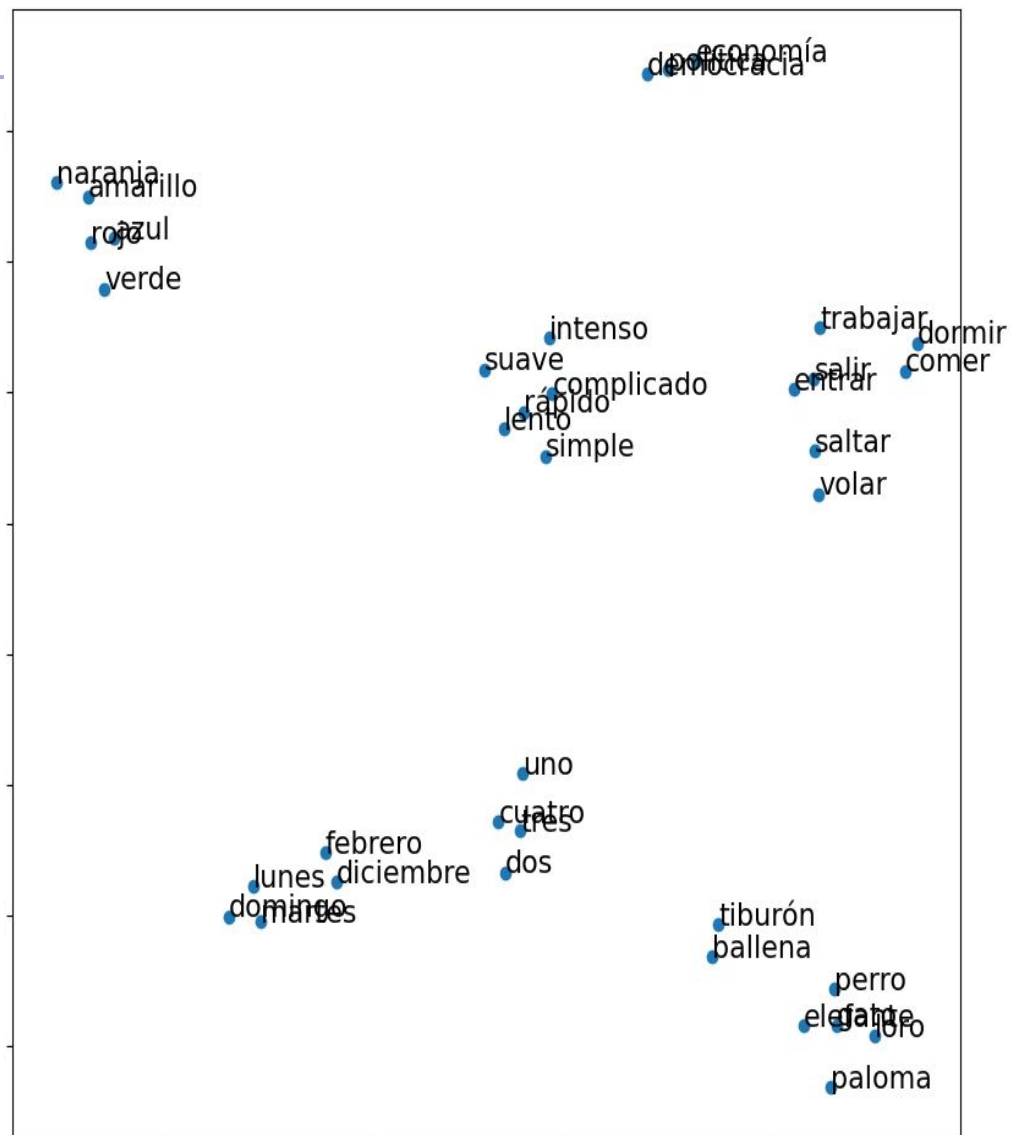
- Distribucionalismo: Lingüistas como Joos (1950), Harris (1954) y Firth (1957) postulan que el significado de las palabras queda definido por cómo se distribuyen en los textos.
- **Palabras que ocurren en contextos similares tienen significados similares.**

Vectores de palabras

Se representa cada palabra con un vector de valores reales.

Palabras similares tendrán vectores cercanos, palabras distintas tendrán vectores lejanos.

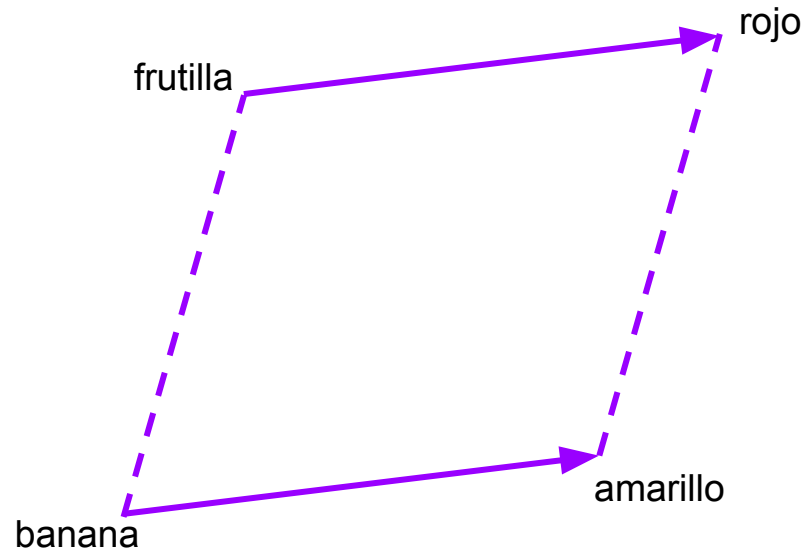
Se procesa muchísimo texto para construir estas representaciones automáticamente (p.e. Mikolov et al., 2013)



Vectores de palabras

Modelo del paralelogramo para razonamientos sobre analogías.

Resolver: “frutilla es a rojo como banana es a _____”



¡Se trasladan muy bien a operaciones con vectores de palabras!

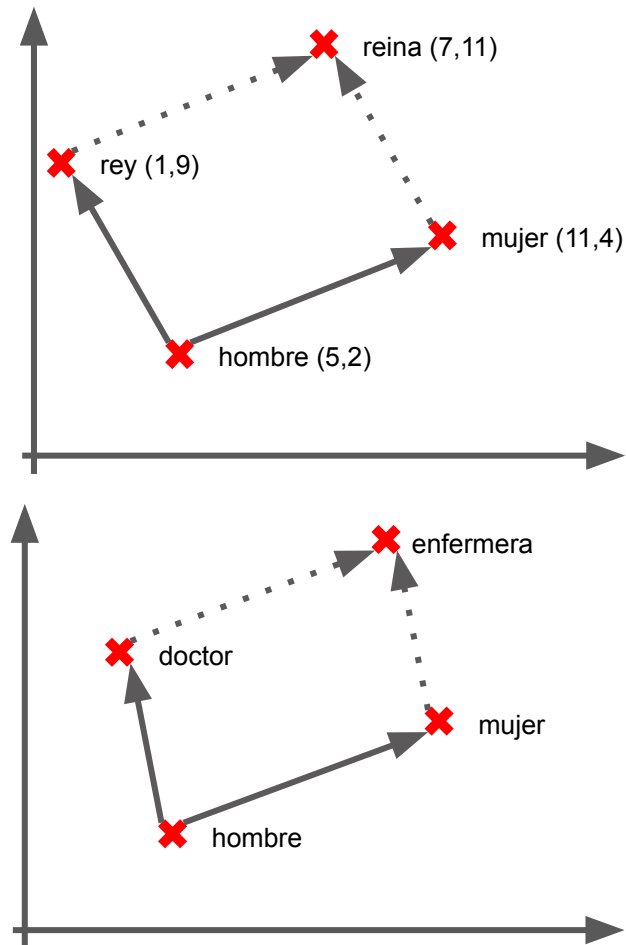
Vectores de palabras

Las operaciones matemáticas sobre vectores pueden descubrir relaciones entre palabras:

rey - hombre + mujer \cong reina

uruguay - montevideo + francia \cong parís

¡Cuidado! También puede amplificar sesgos indeseados incluidos en los datos



Hipótesis distribucional

Hipótesis distribucional

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: $\text{palabra}_N \dots \text{palabra}_2 \text{palabra}_1 X \text{palabra}_1 \text{palabra}_2 \dots \text{palabra}_N$

¿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	
...						

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

Word2Vec

En 2013 Mikolov et al. propusieron **word2vec**: algoritmos para crear colecciones de vectores de palabras **densos** (con pocos 0) y de baja dimensionalidad (por ejemplo, 150 o 300).

Idea: en vez de contar las palabras en una ventana de contexto, entrenamos un clasificador que prediga qué tan probable es que la palabra **c** aparezca en el contexto de **w**.

Como queremos que las palabras **más relacionadas tengan vectores cercanos** y **las menos relacionadas tengan vectores alejados** necesitamos **ejemplos negativos**.

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

Word2Vec

- El objetivo no es usar el clasificador entrenado, sino las representaciones intermedias que se generan dentro de la red neuronal.
- Los pesos aprendidos en la capa oculta de la red son los valores que forman el embedding de la palabra w .
- El entrenamiento es **autosupervisado** porque los valores esperados de salida del clasificador quedan determinados por las palabras que aparecen cerca de w en el texto original (sin anotaciones de ningún tipo).

Word2Vec: Algoritmo skip-gram

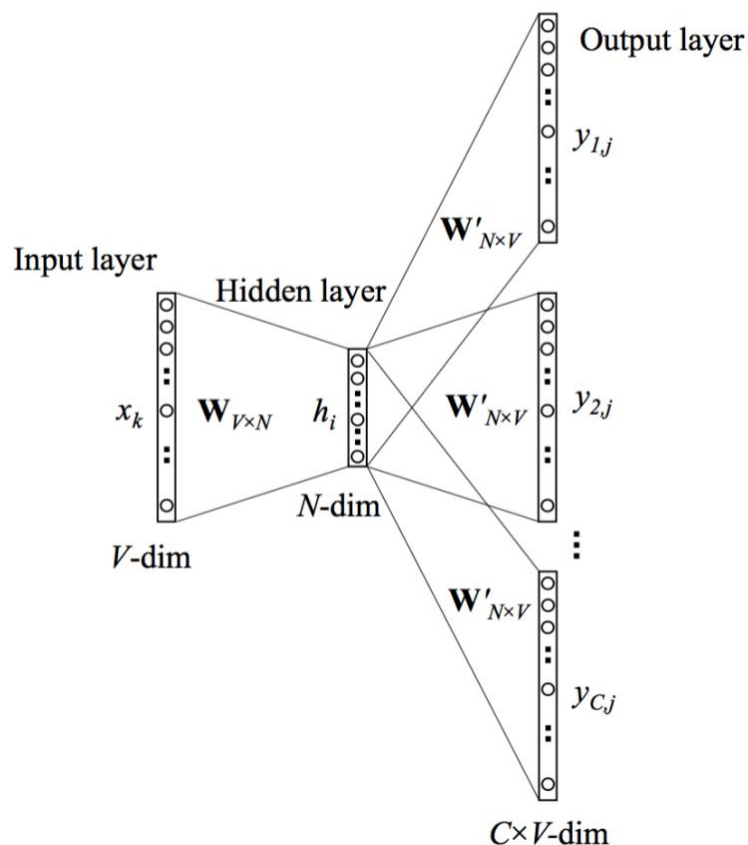


Imagen de "How exactly does word2vec work?"
(Meyer, 2016)

skip-gram intenta modelar las palabras más probables que aparecerán alrededor de una palabra

- **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

Los *word embeddings* son el estado de la capa oculta luego del entrenamiento

Word2Vec: CBOW y skip-gram

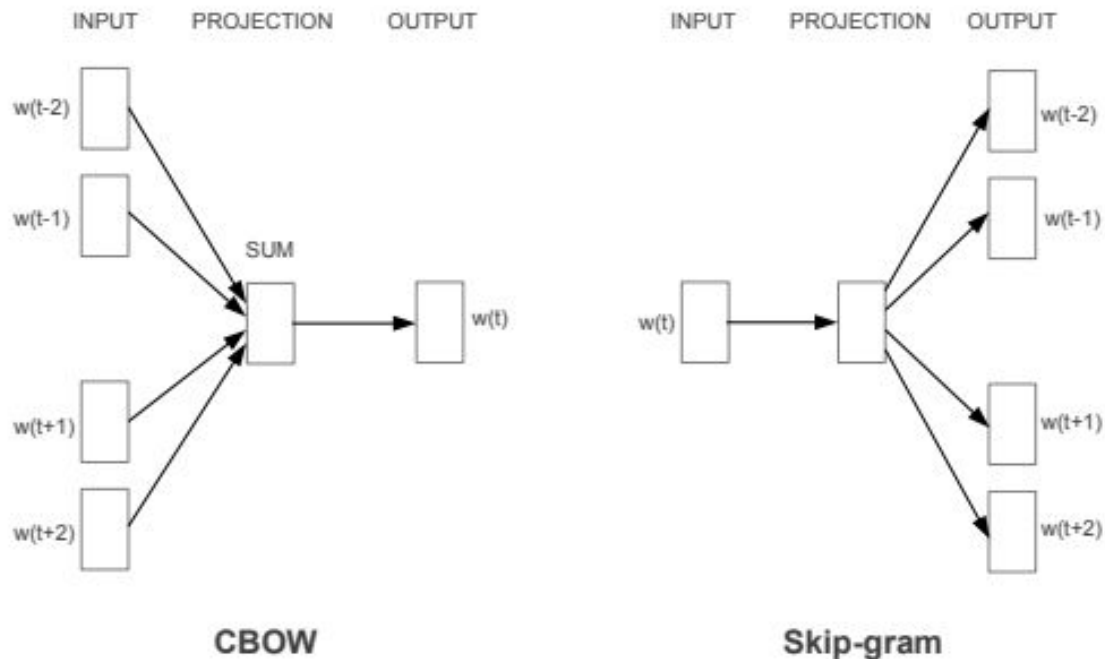




Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781. 2013.

Word2Vec

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

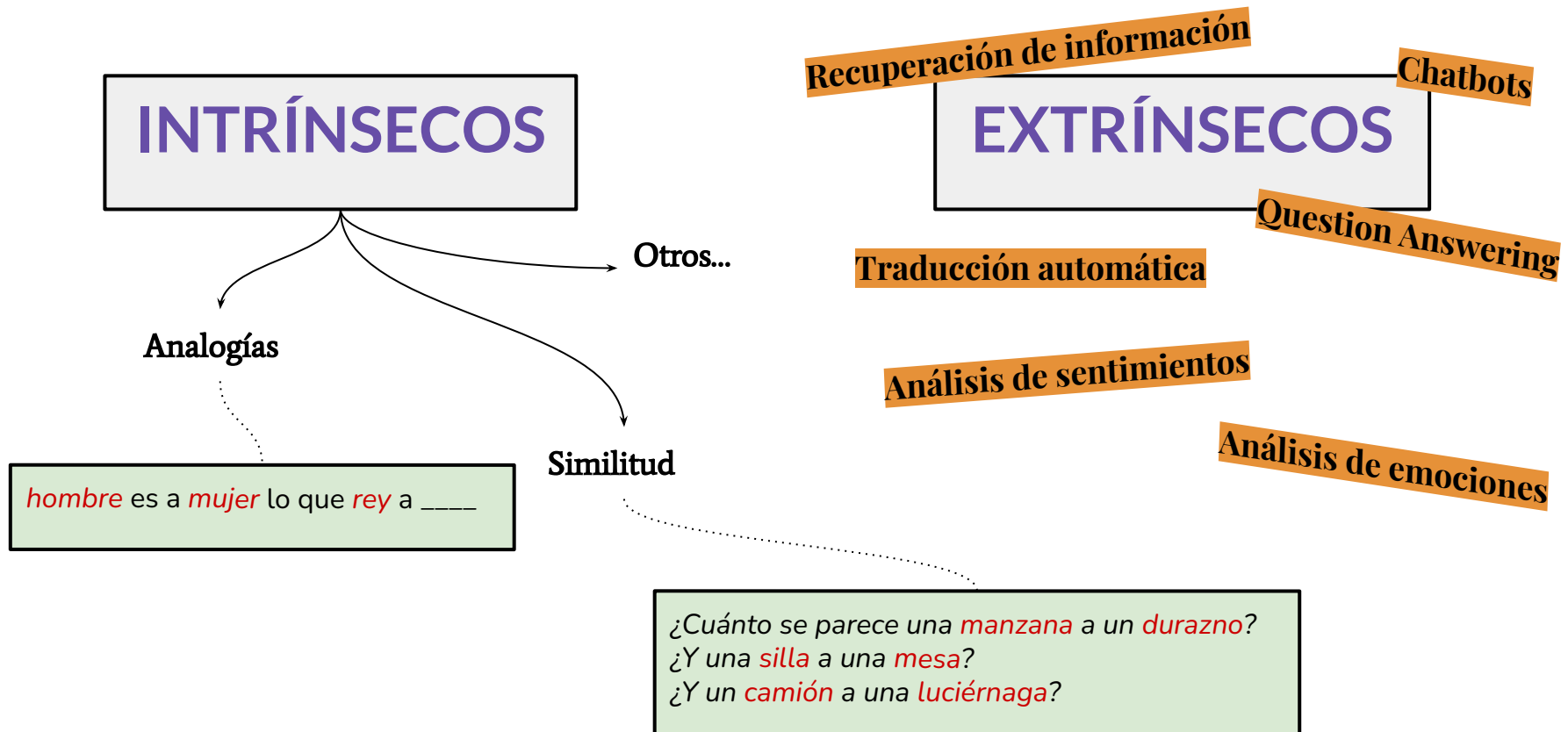
Vectores más cercanos tienden a ser semánticamente similares (similitud coseno).

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

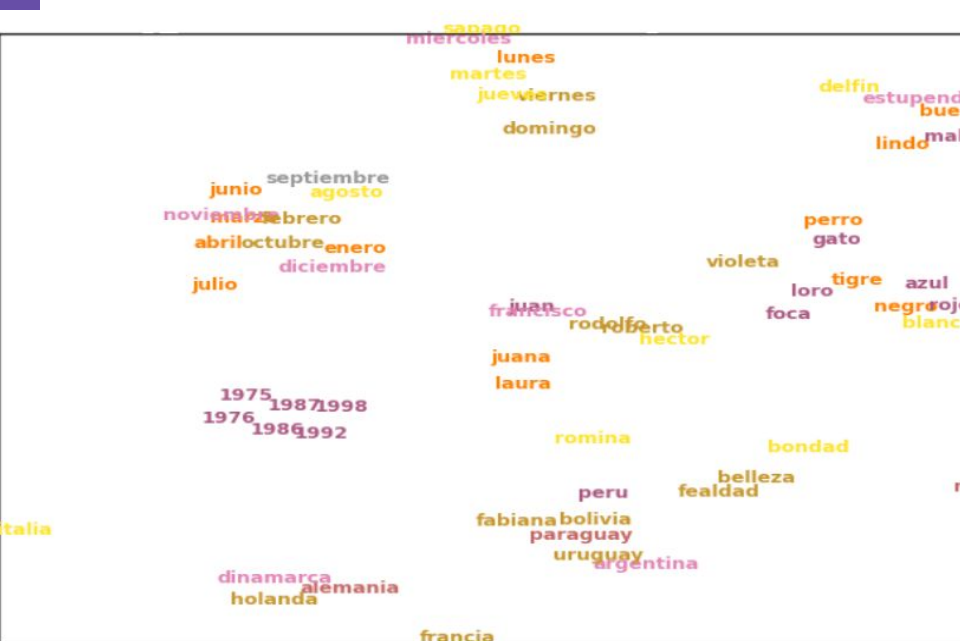
Evaluación

¿Cómo sabemos si una colección de embeddings está bien?

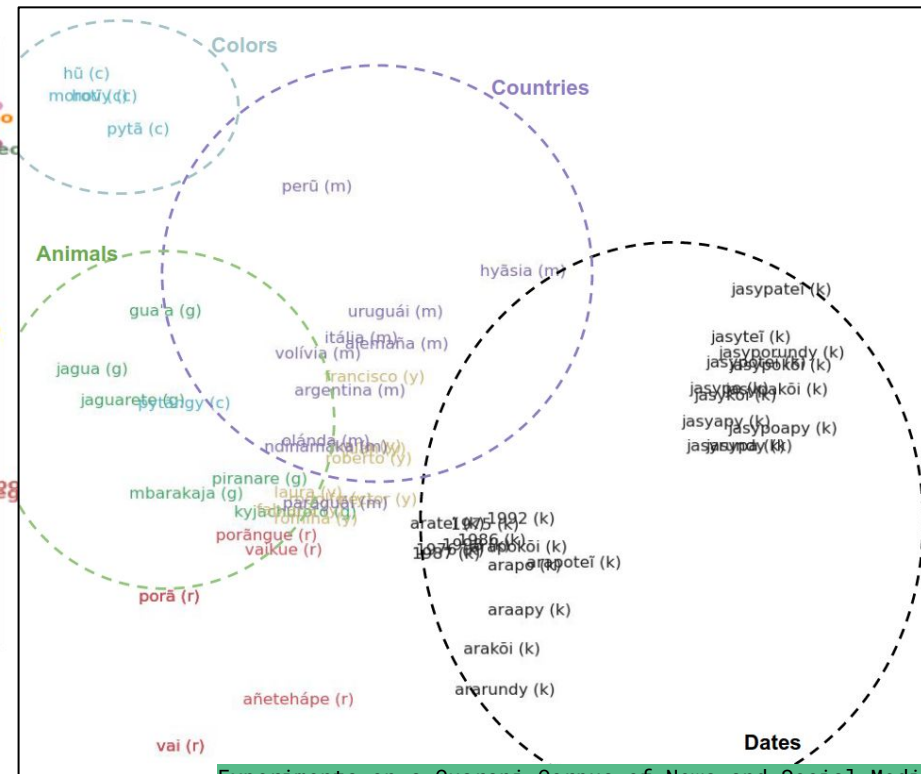


Evaluación

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 Guaraní Corpus of News and Social Media
(Góngora, Giossa & Chiruzzo, AmericasNLP 2021)

Ejemplos

- Conjunto de word embeddings para el español creado por estudiantes de fin de carrera de Ingeniería en Computación ([Azzinari & Martínez, 2015](#))
- Vectores de dimensión 300.
- Corpus de entrenamiento de casi seis mil millones de palabras:

Tipo de documento	Documentos	Palabras	Proporción (palabras)
Noticias	11.318.776	4.376.315.796	73.2 %
Wikipedia	1.113.372	416.932.056	7.0 %
Documentos oficiales	69.965	417.833.686	7.0 %
Escritura amateur	372.627	318.811.674	5.3 %
Libros	7.437	202.546.087	3.4 %
Foros	199.129	133.942.466	2.2 %
Subtítulos	27.105	106.501.897	1.8 %

Cuadro 3.2: Composición del corpus por tipo de documento.

- [Notebook](#)

Embeddings contextuales Sentence embeddings

Los modelos de lenguaje neuronales (que se verán más adelante) dieron lugar a otro tipo de representación vectorial para el lenguaje:

- Embeddings contextuales: se genera un vector diferente para cada palabra en cada contexto.
 - Modelan de mejor manera la ambigüedad.
- Sentence embeddings: representaciones vectoriales para fragmentos de texto (frase, oración, párrafo, documento).
 - Ejemplo

Referencias

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, 3rd edition draft. Stanford. 2024.
[<https://web.stanford.edu/~jurafsky/slp3>
Acceso: setiembre 2024]

Notas del curso Introducción al Procesamiento de Lenguaje Natural (Grupo PLN, Instituto de Computación, Facultad de Ingeniería, Udelar)

[<https://eva.fing.edu.uy/course/view.php?id=211>]

github pln-fing-udelar