X

# Vectorización de palabras

Jocelyn Dunstan Escudero

jdunstan@uc.cl

Departamento de Ciencia de la Computación
& Instituto de Matemática Computacional

Pontificia Universidad Católica de Chile


17 de noviembre de 2025

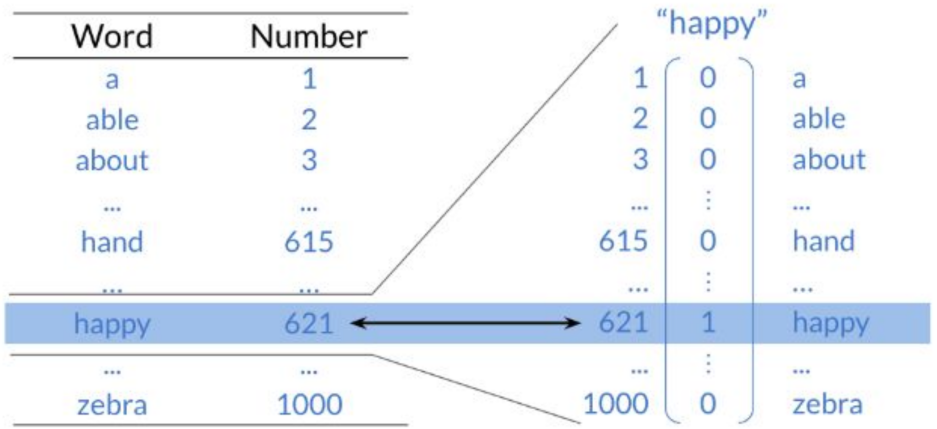
# Objetivos

- Entender por qué es necesario vectorizar las palabras antes de entregarlas a la red neuronal
- Comprender el razonamiento detrás de word2vec

# ¿Cómo le entrego texto a una red neuronal?

# Algunas opciones que tenemos

- Una palabra, un número
- Una palabra, un vector con casi puros ceros menos en una posición (*one-hot enconding)*
- Representación vectorial densa (*word embeddings* de dimensión 50-500




Extraído desde NLP with Probabilistic Models desde Coursera

# ¿Qué propiedades nos gustaría que tuvieran estos vectores?

- Hay palabras que tienen significados similares (sofá, sillón) y otras que aparecen en contextos similares (vaca, caballo)

# ¿Qué propiedades nos gustaría que tuvieran estos vectores?

- Hay palabras que tienen significados similares (sofá, sillón) y otras que aparecen en contextos similares (vaca, caballo)

- Existen palabras polisémicas (e.g. banco)

# ¿Qué propiedades nos gustaría que tuvieran estos vectores?

- Hay palabras que tienen significados similares (sofá, sillón) y otras que aparecen en contextos similares (vaca, caballo)
- Existen palabras polisémicas (e.g. banco)
- Existe el principio lingüístico del contraste: diferencia en la forma implica diferencia en el significado

# ¿Qué propiedades nos gustaría que tuvieran estos vectores?

- Hay palabras que tienen significados similares (sofá, sillón) y otras que aparecen en contextos similares (vaca, caballo)
- Existen palabras polisémicas (e.g. banco)
- Existe el principio lingüístico del contraste: diferencia en la forma implica diferencia en el significado
- "El significado de una palabra es su uso en el lenguaje"(Ludwig Wittegenstein)

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf


https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Hipótesis distribucional

"Palabras que ocurren en contextos similares tienden a tener significados similares" (Joos, Harris, Firth, **1950s**)

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Word2vec: Idea clave

Podemos usar el texto que ya tenemos para plantearlo como un problema de aprendizaje supervisado: **predecir la palabra del medio dado el contexto.**

# Le damos ejemplos positivos y negativos

...lemon, a [tablespoon of apricot jam, a] pinch...

c1 c2 [target] c3 c4

**positive examples +**

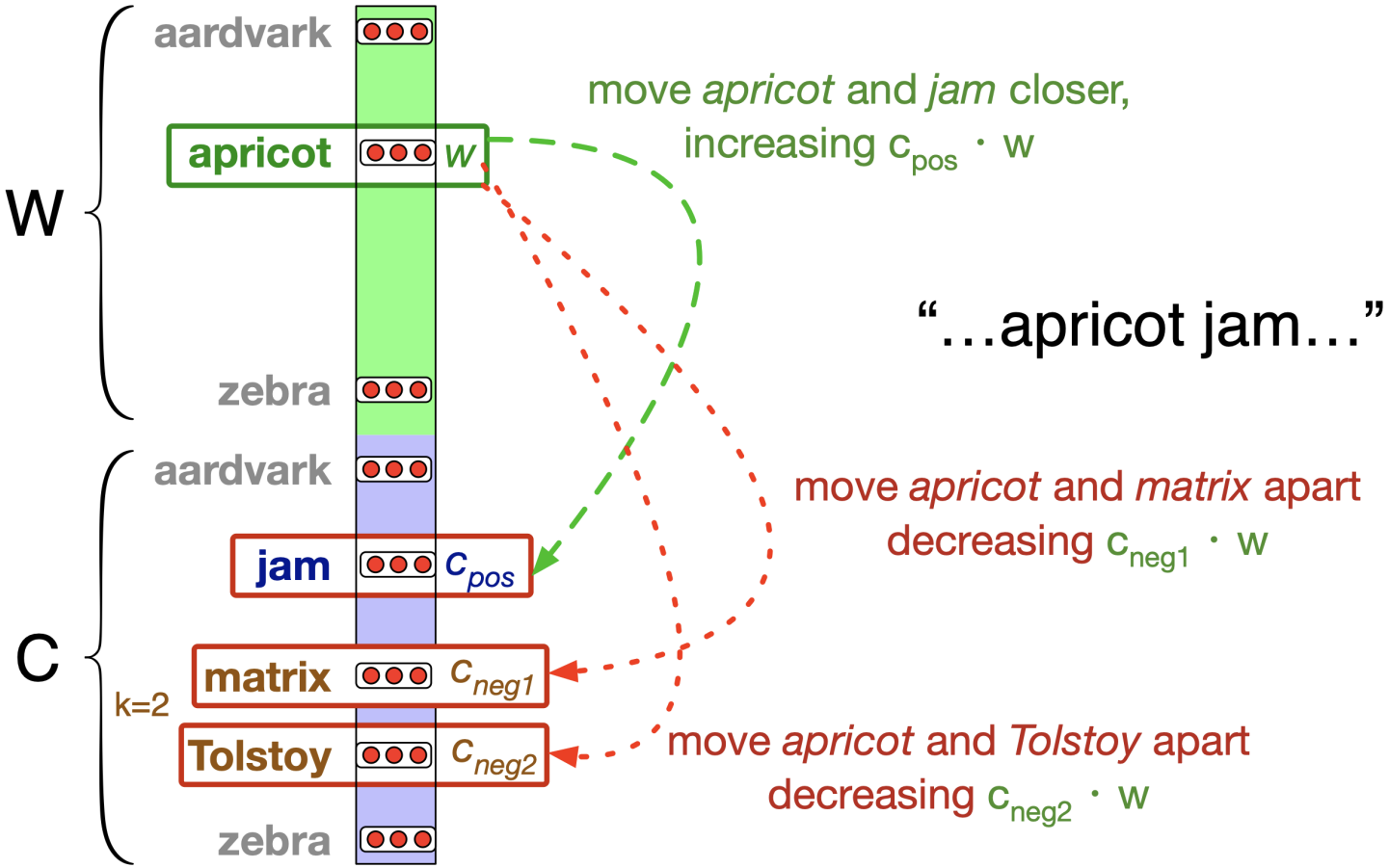
| t | c |
|---|---|
| apricot | tablespoon |
| apricot | of |
| apricot | jam |
| apricot | a |

**negative examples -**

| t | c | t | c |
|---|---|---|---|
| apricot | aardvark | apricot | seven |
| apricot | my | apricot | forever |
| apricot | where | apricot | dear |
| apricot | coaxial | apricot | if |

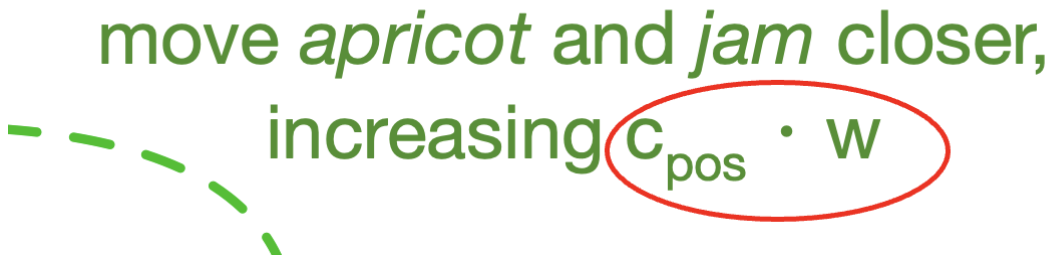
https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Partimos de vectores aleatorios y vamos ajustando con decenso del gradiente




https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

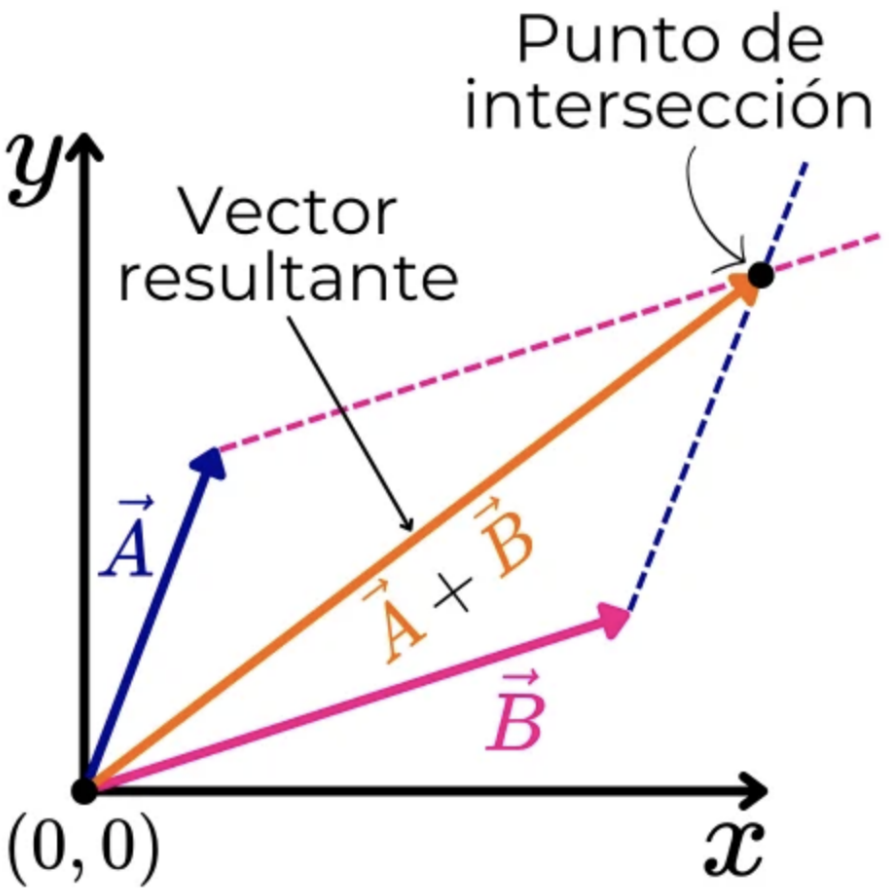
# Y calculamos la similaridad entre palabras como el producto punto entre vectores




https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

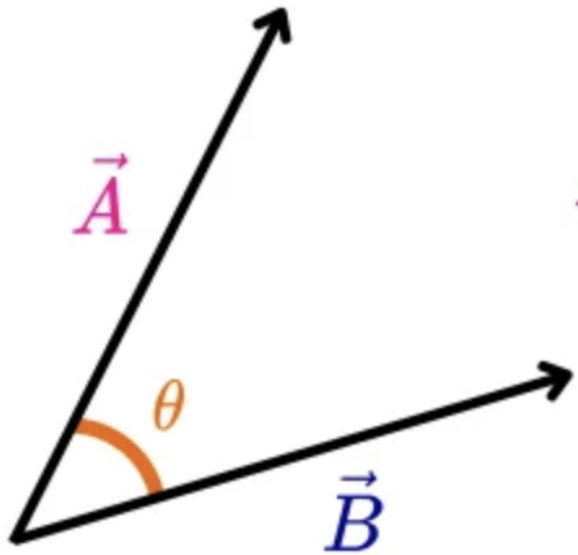
# Repaso de vectores

# ¿Cómo se suman vectores?




https://dicciomat.com/tipos-de-vectores/

# ¿Qué es el producto punto entre vectores?

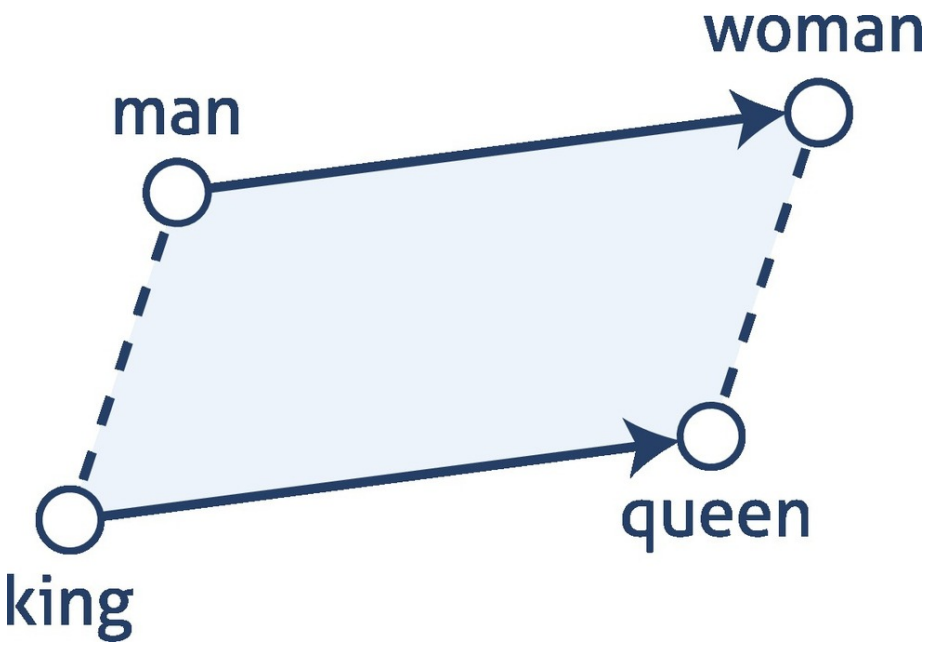



$$\vec{A} \cdot \vec{B} = \left|\vec{A}\right| \left|\vec{B}\right| \cos\theta$$
$$= ab\cos\theta$$

https://dicciomat.com/producto-punto-de-vectores/

# Relaciones analógicas

# Relaciones analógicas




# Pueden usarse para medir sesgo

Ask “Paris : France :: Tokyo : x”

- x = Japan

Ask “father : doctor :: mother : x”

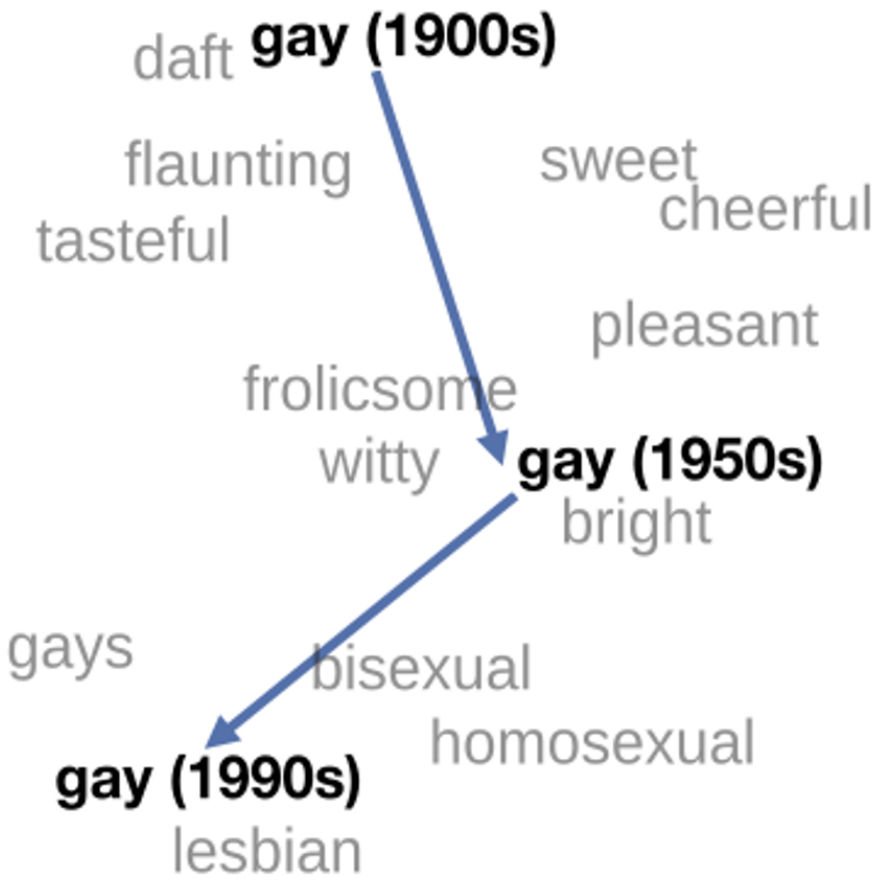
- x = nurse

Ask “man : computer programmer :: woman : x”

- x = homemaker

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Una mirada al significado histórico de palabras




https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# ¿Cómo se aplica descenso del gradiente en este contexto?

# Sabemos que la similaridad de palabras se mide usando producto punto..

$$Similarity(w, c) \approx \mathbf{c} \cdot \mathbf{w}$$

# Sabemos que la similaridad de palabras se mide usando producto punto..

$$Similarity(w, c) \approx \mathbf{c} \cdot \mathbf{w}$$

Pero esto no es una probabilidad. Como es usual en IA usamos la función sigmoide:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Las probabilidades de ejemplos positivos y negativos es:

$$P(+|w,c) = \sigma(\mathbf{c}\cdot\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{c}\cdot\mathbf{w})}$$

# Las probabilidades de ejemplos positivos y negativos es:

$$P(+|w,c) = \sigma(\mathbf{c}\cdot\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{c}\cdot\mathbf{w})}$$

$$\begin{aligned} P(-|w,c) &= 1 - P(+|w,c) \\ &= \sigma(-\mathbf{c}\cdot\mathbf{w}) = \frac{1}{1+\exp(\mathbf{c}\cdot\mathbf{w})} \end{aligned}$$

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Y por lo tanto, la función de pérdida es:

$$
\begin{aligned}
L &= -\log\left[P(+|w,c_{pos})\prod_{i=1}^{k}P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos}) + \sum_{i=1}^{k}\log P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos}) + \sum_{i=1}^{k}\log\left(1-P(+|w,c_{neg_i})\right)\right] \\
&= -\left[\log \sigma(c_{pos}\cdot w) + \sum_{i=1}^{k}\log\sigma(-c_{neg_i}\cdot w)\right]
\end{aligned}
$$

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Cuya derivada puede demostrarse que es:

$$\frac{\partial L}{\partial c_{pos}} = [\sigma(\mathbf{c}_{pos} \cdot \mathbf{w}) - 1]\mathbf{w}$$

$$\frac{\partial L}{\partial c_{neg}} = [\sigma(\mathbf{c}_{neg} \cdot \mathbf{w})]\mathbf{w}$$

$$\frac{\partial L}{\partial w} = [\sigma(\mathbf{c}_{pos} \cdot \mathbf{w}) - 1]\mathbf{c}_{pos} + \sum_{i=1}^{k} [\sigma(\mathbf{c}_{neg_i} \cdot \mathbf{w})]\mathbf{c}_{neg_i}$$

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

# Por lo tanto, el descenso del gradiente es:

$$
\begin{aligned}
\mathbf{c}_{pos}^{t+1} &= \mathbf{c}_{pos}^{t} - \eta[\sigma(\mathbf{c}_{pos}^{t} \cdot \mathbf{w}^{t}) - 1]\mathbf{w}^{t} \\
\mathbf{c}_{neg}^{t+1} &= \mathbf{c}_{neg}^{t} - \eta[\sigma(\mathbf{c}_{neg}^{t} \cdot \mathbf{w}^{t})]\mathbf{w}^{t} \\
\mathbf{w}^{t+1} &= \mathbf{w}^{t} - \eta\left[[\sigma(\mathbf{c}_{pos}^{t} \cdot \mathbf{w}^{t}) - 1]\mathbf{c}_{pos}^{t} + \sum_{i=1}^{k}[\sigma(\mathbf{c}_{neg_i}^{t} \cdot \mathbf{w}^{t})]\mathbf{c}_{neg_i}^{t}\right]
\end{aligned}
$$

Donde $\eta$ es la tasa de aprendizaje.

https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf