Día 08 · Tema 3 — Embeddings: word2vec, distancias y vectores contextuales

Eduardo C. Garrido-Merchán · ecgarrido@comillas.edu Universidad Pontificia Comillas · ICADE

1. Qué es un embedding

Un *embedding* es una función $\phi : \mathcal{V} \to \mathbb{R}^d$ que asocia a cada token (palabra, frase, párrafo, imagen, audio…) de un vocabulario $\mathcal{V}$ un vector denso de dimensión d (típicamente $d \in [128, 4096]$). El criterio que organiza el espacio es: tokens semánticamente cercanos $\Rightarrow$ vectores cercanos.

2. La aritmética semántica de word2vec

El resultado canónico es

$$\phi(\text{rey}) \;-\; \phi(\text{hombre}) \;+\; \phi(\text{mujer}) \;\approx\; \phi(\text{reina}). \tag{1}$$

La operación tiene sentido porque `rey` y `reina` comparten el componente "realeza" y difieren en el componente "género"; restar `hombre` y sumar `mujer` es un *paseo* por el eje de género en el espacio de embeddings.

3. Distancia coseno

La similitud estándar entre vectores de embeddings es la coseno:

$$\cos(u, v) \;=\; \frac{u^\top v}{\|u\|_2 \, \|v\|_2} \;=\; \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\,\sqrt{\sum_i v_i^2}} \in [-1, 1]. \tag{2}$$

$u, v \in \mathbb{R}^d$	los dos vectores comparados
$u^\top v$	producto escalar
$\|u\|_2$	norma euclídea

La distancia coseno es $d_{\cos} = 1 - \cos(u, v) \in [0, 2]$. **Por qué coseno y no euclídea**: en alta dimensión las normas se concentran y la distancia euclídea pierde discriminación; el ángulo (dirección) sigue siendo informativo.

3.1. Ejemplo a mano

$u = (1, 2, 3)$, $v = (2, 4, 6)$: $u \cdot v = 2 + 8 + 18 = 28$, $\|u\| = \sqrt{14} \approx 3{,}74$, $\|v\| = \sqrt{56} \approx 7{,}48$, $\cos(u, v) = 28/(3{,}74 \cdot 7{,}48) = 1{,}00$. Son colineales: representan lo mismo en distinta escala.

4. Cómo se entrenan: el objetivo skip-gram

Para una palabra central w_t y una ventana de contexto C_t, el modelo maximiza la log-verosimilitud

$$\mathcal{L}(\theta) \;=\; \sum_{t=1}^{T} \sum_{w_c \in C_t} \log p_\theta(w_c \mid w_t), \tag{3}$$

con $p_\theta(w_c|w_t) = \text{softmax}\big(\phi(w_c)^\top \phi(w_t)\big)$. Como el softmax sobre $|\mathcal{V}| \sim 10^5$–10^6 es caro, se usa *negative sampling*: maximizar

$$\log \sigma\big(\phi(w_c)^\top \phi(w_t)\big) \;+\; \sum_{j=1}^{k} \mathbb{E}_{w_n \sim P_n}\Big[\log \sigma\big(-\phi(w_n)^\top \phi(w_t)\big)\Big]. \tag{4}$$

Intuitivamente: *acercar* el embedding del par real (w_t, w_c) y *alejar* el de k palabras muestreadas al azar del vocabulario.

5. Embeddings contextuales (BERT, MPNet)

En word2vec un token tiene *un* vector independientemente del contexto: `banco` significa lo mismo en "ir al banco" que en "sentarse en el banco". Los embeddings contextuales producen $\phi(w_t \mid w_{1:T})$, que depende de la frase entera:

$$\phi_{\text{ctx}}(w_t) \;=\; \text{Transformer}\big(w_{1:T}\big)_t, \tag{5}$$

y son distintos en distintas frases. SBERT y MPNet son los modelos estándar para producir embeddings de oración: pasan la frase entera por un encoder y aplican *mean pooling* sobre las posiciones.

6. Hitos de la década

- **2013**: word2vec (Mikolov), GloVe (Pennington). $d = 300$.
- **2018**: ELMo y BERT — embeddings contextuales bidireccionales.
- **2019**: SBERT — embeddings de oración con *Siamese networks*.
- **2022**: OpenAI `text-embedding-ada-002`, $d = 1536$.
- **2024–26**: `text-embedding-3-large` ($d = 3072$), BGE, E5, Cohere Embed v3, Mistral Embed — multilingüe.

7. Aplicación a búsqueda semántica

Tres pasos: **(1)** indexar el corpus: para cada documento d_j calcular $\phi(d_j)$ y guardarlo en una base vectorial (FAISS, Milvus, Qdrant). **(2)** para una query q, calcular $\phi(q)$. **(3)** recuperar los k documentos con mayor $\cos(\phi(q), \phi(d_j))$. Esta es la columna vertebral de RAG (Día 13).

Idea clave. El criterio práctico para elegir un modelo de embedding es: *(latencia objetivo)* → *tamaño en MB; (idioma)* → *multilingüe o no; (dominio)* → *general o domain-tuned.* Para castellano legal, `intfloat/multilingual-e5-large` o BGE-M3 son la elección defensiva.