Deep Learning para Business Analytics

Día 08 · Tema 3 — Embeddings semánticos

Madrid, Día 08

📚 *Vectores que significan.*

Eduardo C. Garrido-Merchán

Profesor Propio Adjunto, Métodos Cuantitativos, ICADE

ecgarrido@comillas.edu

BBVA · MERCADONA · NETFLIX · amazon · Spotify · Rakuten

ejemplos del día

COMILLAS UNIVERSIDAD PONTIFICIA ICAI ICADE CIHS

Lo que vamos a cubrir hoy.

1. Qué es un embedding

2. Cómo se entrenan

3. Estado del arte

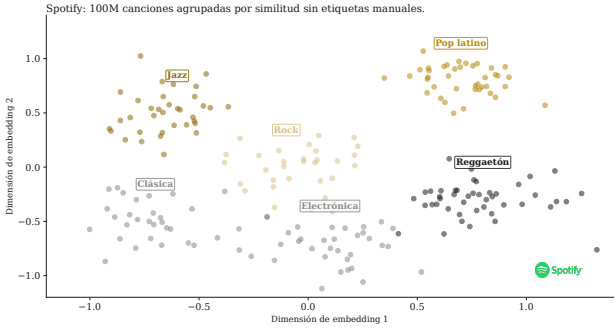
4. Casos de negocio

Al final del día: sabes calcular distancias coseno entre embeddings 📚 y entiendes la base de RAG 🔍 (Tema 6).

Parte 1 / 4

Qué es un embedding

🎵🔍 Spotify agrupa 100 millones de canciones por similitud sin etiquetas manuales.



El problema. Spotify tiene > 100 millones de canciones. Etiquetarlas a mano es imposible.

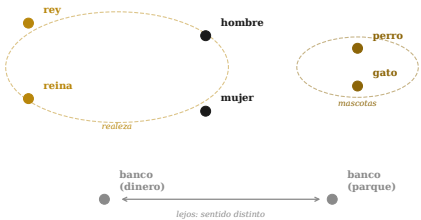
La solución. Un embedding convierte cada canción en un vector. Canciones parecidas (por audio, letras y comportamiento de escucha) quedan como vectores cercanos.

El resultado. Discover Weekly y el autoplay funcionan comparando la distancia entre tu historial y canciones candidatas en el espacio de embeddings.

Idea clave. Un embedding convierte datos complejos en vectores donde la cercanía es similitud.

🌍 Un embedding es una dirección postal para cada palabra.

Un embedding es un mapa semántico: palabras cercanas = significados cercanos.



El mapa semántico. Cada palabra vive en un punto del espacio. Palabras con significado parecido (rey/reina, perro/gato) están cerca; palabras con sentido distinto (banco-dinero vs. banco-parque) están lejos.

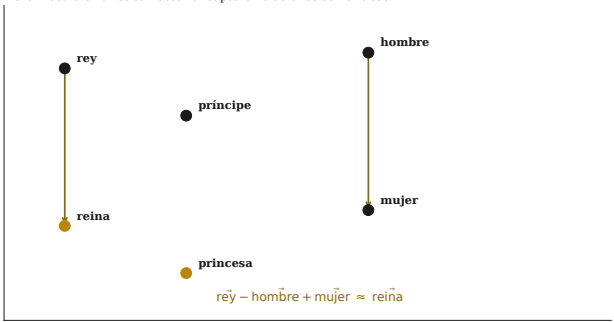
La métrica. La distancia coseno mide el ángulo entre dos vectores. Ángulo pequeño → significado compartido. Ángulo grande → sentidos distintos.

Por qué importa. Esta geometría permite buscar por significado, no por coincidencia de caracteres.

Idea clave. Palabras con significado parecido viven cerca. La distancia coseno mide cuánto.

word2vec 🧠: la aritmética vectorial captura significado.

word2vec: la aritmética vectorial captura relaciones semánticas.



La analogía. 👸 rey – 👨 hombre + 👩 mujer ≈ 👸 reina.

Lectura. El vector de 👸 rey a 👸 reina es paralelo al de 👨 hombre a 👩 mujer: la dirección codifica género.

Por qué importa. Si la aritmética funciona, los vectores han capturado relaciones semánticas reales, no solo co-ocurrencia.

Ejemplo a mano ⚙️: distancia coseno entre vectores.

Distancia coseno entre embeddings: el sentido emerge del entorno.

Palabra	Vector 3D	Distancia coseno a "banco financiero"
banco financiero	$(0.8,\ 0.5,\ 0.3)$	1,00 (consigo misma)
entidad bancaria	$(0.7,\ 0.6,\ 0.4)$	0,99
hipoteca	$(0.6,\ 0.7,\ 0.4)$	0,96
banco de parque	$(-0.1,\ -0.4,\ 0.9)$	0,29
árbol	$(-0.5,\ -0.6,\ 0.7)$	-0,12

La distancia coseno en una fórmula 🎯, sin trampa.

$$\cos(\vec{u}, \vec{v}) \;=\; \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \;\in\; [-1, 1].$$

⚖️ **Por qué coseno y no euclidiana.** Los embeddings se comparan por dirección, no por magnitud. Vectores cortos y largos en la misma dirección son "lo mismo" semánticamente.

⚠️ **Cuándo dudar.** Si los vectores no están normalizados a norma 1 y la magnitud sí codifica algo (por ejemplo, frecuencia), entonces euclidiana puede tener sentido.

Idea clave. 🔍 En RAG, sentence-similarity y búsquedas semánticas: usa coseno y normaliza al guardar.

Parte 2 / 4

Cómo se entrenan

Skip-gram 🕸 (word2vec): predice el contexto desde la palabra.

Skip-gram: dado el token central, predecir las palabras de contexto.

⇒ el embedding de "banco" acumula su contexto.

⚙️ **Mecanismo.** Dada la palabra central, el modelo predice las palabras vecinas en una ventana de contexto.

📚 **Resultado.** La capa oculta, tras el entrenamiento, contiene los vectores-embedding de cada palabra del vocabulario.

🚀 **Escala.** Google entrenó el word2vec original sobre 100 000 millones de palabras de Google News.

🎯 Distancia coseno a mano entre una query y tres documentos.

Distancia coseno a mano entre una query y tres documentos.

Vectores	$q = (1, 0, 1)$; $d_1 = (1, 1, 0)$; $d_2 = (0, 0, 1)$; $d_3 = (1, 0, 1)$
$\|q\|$	$= \sqrt{1^2 + 0^2 + 1^2} = \sqrt{2} \approx 1.414$
$\cos(q, d_1)$	$= \frac{1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0}{\sqrt{2} \cdot \sqrt{2}} = \frac{1}{2} = 0.500$
$\cos(q, d_2)$	$= \frac{1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1}{\sqrt{2} \cdot 1} = \frac{1}{\sqrt{2}} = 0.707$
$\cos(q, d_3)$	$= \frac{1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1}{\sqrt{2} \cdot \sqrt{2}} = \frac{2}{2} = 1.000$
Ranking	d_3 (1.000) > d_2 (0.707) > d_1 (0.500)

Setup. $q = (1, 0, 1)$; $d_1 = (1, 1, 0)$; $d_2 = (0, 0, 1)$; $d_3 = (1, 0, 1)$.

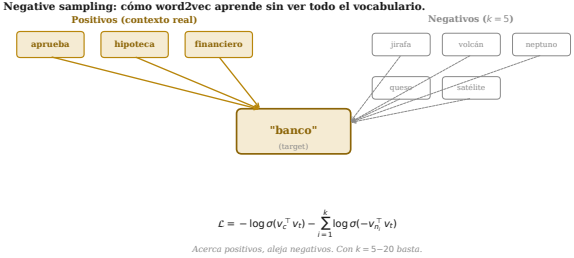
Normas. $\|q\| = \sqrt{2}$; $\|d_1\| = \sqrt{2}$; $\|d_2\| = 1$; $\|d_3\| = \sqrt{2}$.

Cosenos. $\cos(q, d_1) = 1/2 = 0{,}500$.
$\cos(q, d_2) = 1/\sqrt{2} = 0{,}707$.
$\cos(q, d_3) = 2/2 = 1{,}000$.

Ranking. d_3 (1.000) > d_2 (0.707) > d_1 (0.500). El documento d_3 es idéntico en dirección a la query.

Idea clave. Verifica: $d_3 = q$ exactamente, así que cos = 1. Este es el principio de la búsqueda semántica.

🎯🧠 Negative sampling: cómo word2vec aprende sin ver todo el vocabulario.



El problema. Softmax sobre $|V| = 100\,000$ palabras es prohibitivo en cada paso de entrenamiento.

La idea. Entrena con $k = 5$–20 "negativos" al azar. Acerca el target a sus positivos y aléjalo de los negativos.

La pérdida. $\mathcal{L} = -\log \sigma(v_c^\top v_t) - \sum_{i=1}^{k} \log \sigma(-v_{n_i}^\top v_t)$

Conexión. Negative sampling aproxima la estimación contrastiva de ruido (NCE).

Idea clave. Con 5–20 negativos, word2vec aprende de embeddings de calidad sobre vocabularios enormes.

Embeddings contextuales 🎭: la misma palabra cambia de vector.

En word2vec. "🏦 Banco" tiene un *único* vector que mezcla todos sus sentidos (financiero y de parque).

En BERT 🎭 y descendientes. "Banco" produce un vector distinto según el contexto. "El *banco* aprueba la hipoteca" 💰 → vector cercano a "entidad financiera". "Me senté en el *banco* del parque" 🌳 → vector cercano a "asiento".

⚙️ **Cómo se entrena.** Masked Language Modeling: ocultas el 15 % de tokens y la red aprende a reconstruirlos viendo todo el contexto bidireccional.

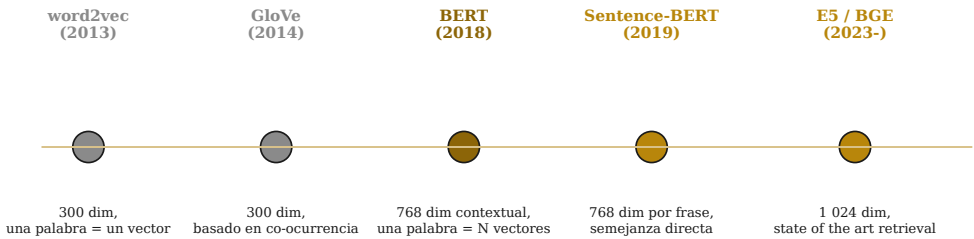
Idea clave. 🔍 Embeddings contextuales son la pieza fundamental de la búsqueda semántica empresarial moderna.

Parte 3 / 4

Estado del arte

⏱️ Una década de embeddings, en cinco hitos.

Una década de embeddings, condensada en cinco hitos.



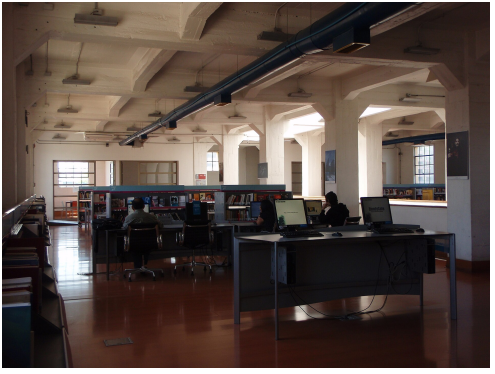
📊 Concepto → aplicación: qué embedding para qué tarea.

Modelo	Caso	Entrada / salida	Dim
🧠 word2vec / GloVe	Buscador interno simple, NLP clásico.	Palabra → vector estático.	300
🎭 BERT base	Clasificación de sentimiento, NER.	Frase → N vectores (uno/token).	768
🔍 Sentence-BERT	Búsqueda semántica de párrafos.	Párrafo → 1 vector.	768
🚀 E5 / BGE large	RAG corporativo (Día 15).	Pregunta o doc → 1 vector.	1024
👁️ CLIP	Búsqueda imagen-texto.	Imagen o texto → vector compartido.	512

Idea clave. 🤖 Hugging Face publica modelos con tarjeta de licencia y benchmark MTEB.

Parte 4 / 4

Casos de negocio



⭐ Cuatro casos que viven sobre embeddings.

Cuatro casos de negocio que viven sobre embeddings.

🔍 **Buscador semántico.** Encuentra documentos por significado, no por palabra exacta.

🛒 **Recomendador.** Productos similares en el espacio de embeddings.

🏦 **Detección de fraude.** Transacciones anómalas como vectores lejanos del clúster normal.

🏥 **Triaje clínico.** Clasificar urgencias por similitud semántica con casos previos.

🔧 Receta exprés: similitud semántica en 10 líneas.

```
from sentence_transformers import SentenceTransformer
import numpy as np

m = SentenceTransformer(ïntfloat/multilingual-e5-large")
docs = ["la hipoteca incluye comisión de apertura",
    .el préstamo personal tiene tipo fijo",
    .el banco del parque es de madera"]
embs = m.encode(docs, normalize_embeddings=True)

query = m.encode("¿cuáles son los costes de una hipoteca?",
    normalize_embeddings=True)
sims = embs @ query # producto-punto = coseno (normalizado)
print(np.argsort(-sims)) # orden de relevancia
```

Idea clave. 🚀 Estas 10 líneas son la base del Día 15 (RAG corporativo).

🎓 Lo que el alumno se lleva a casa de este día.

📚 **Concepto 1.** Un embedding es un vector denso que codifica significado: la cercanía geométrica refleja cercanía semántica.

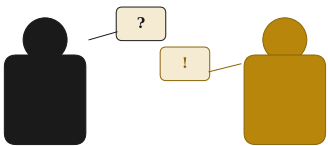
🎭 **Concepto 2.** El estado del arte ha pasado de estáticos (word2vec) a contextuales (BERT) y a especializados para retrieval (E5/BGE).

🎯 **Concepto 3.** Distancia coseno + normalización son la combinación canónica.

🔧 **Para la práctica.** Embedder gratis (E5) + buscador semántico sobre 100 párrafos corporativos.

⚡ **Próximo día.** Día 09 (Tema 3.2): atención multi-head y bloque Transformer completo.

💬 Pausa para debatir: ¿Codifica un embedding también los prejuicios?



⚠️ **La pregunta.** Un embedding mide significado, pero ¿de quién? ¿Heredan los prejuicios del corpus de entrenamiento?

🔥 **Posición A.** Sí, y los amplifican: “médico” vs. “enfermera” en word2vec lo mostró.

🛡️ **Posición B.** No, son herramientas neutras; el problema es el uso, no el embedding.

⏱️ **Reglas.** 5 minutos, dos turnos de palabra por bando, móviles boca abajo. Yo modero, no participo.

Hasta el Día 09.

⚡ *El Transformer entero, en pizarra.*

Eduardo C. Garrido-Merchán

ecgarrido@comillas.edu