Deep Learning para Business Analytics

Día 15 · Tema 6 — RAG corporativo

Conocimiento privado con citas verificables.

Eduardo C. Garrido-Merchán

Profesor Propio Adjunto, Métodos Cuantitativos, ICADE

ecgarrido@comillas.edu

Madrid, Día 15

ejemplos del día

Lo que vamos a cubrir hoy.

1. Por qué RAG

2. Pipeline RAG canónico

3. Patrones de retrieval

4. Caso BBVA: dimensiones reales

Al final del día: montas un RAG mínimo sobre tus propios documentos y argumentas por qué RAG suele ganar a fine-tuning.

Parte 1 / 4

Por qué RAG

El problema del conocimiento privado.

Realidad. Un LLM 🧠 se entrena con datos hasta una fecha de corte. No conoce tu CRM, tus circulares ni el calendario de la oficina de Burgos.

Opción A — Fine-tuning. Entrenar el modelo con tus datos. Caro, lento, mantenimiento continuo (cada nuevo doc requiere reentrenar).

Opción B — RAG. Mantener los documentos 📄 en una base vectorial. En cada consulta, recuperar los más relevantes 🔍 y pasarlos como contexto. Documento nuevo ⇒ se indexa solo.

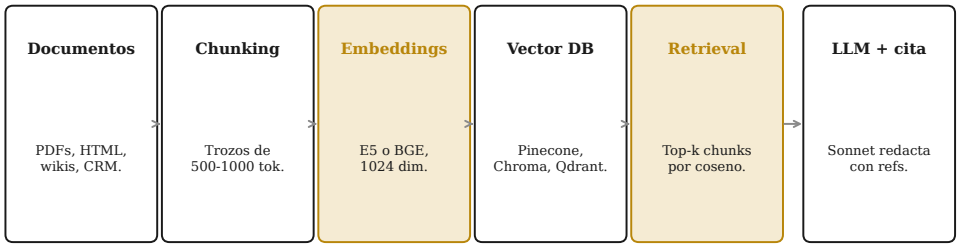
Idea clave. Para conocimiento corporativo cambiante, RAG es la respuesta correcta el 90 % de las veces.

Parte 2 / 4

Pipeline RAG canónico

Pipeline RAG 📚: ingesta una vez, retrieval en cada query.

Pipeline RAG canónico: ingesta una vez, retrieval en cada query.



Chunking 📄: tres estrategias y cuándo usarlas.

Tres estrategias de chunking: fixed, sentence, recursive.

Fixed size

Corta cada N tokens sin importar el contenido. Simple pero rompe oraciones a la mitad.

Ideal para:

Logs, datos tabulares.

Sentence-aware

Corta en fronteras de oración. Respeta la estructura del texto pero chunks desiguales.

Ideal para:

Artículos, emails, FAQs.

Recursive + overlap

Divide jerárquicamente (sección > párrafo > oración) con solapamiento del 10-20 %.

Ideal para:

Documentos largos (contratos, circulares).

Concepto → aplicación: qué bases vectoriales.

Vector DB	Caso	Coste / Ops	Licencia
Pinecone	RAG empresarial gestionado.	Cloud, $100–3 000/mes.	Propietaria.
Qdrant	RAG self-hosted production.	Servidor propio.	Apache 2.0.
Chroma	RAG dev / prototipos.	Local, en memoria.	Apache 2.0.
Weaviate	RAG híbrido con texto + vector.	Cloud o self-hosted.	BSD-3.
pgvector	RAG pequeño dentro de Postgres.	Reusa BBDD existente.	PostgreSQL.

Idea clave. Empieza con Chroma local; sube a Qdrant o Pinecone cuando crezca el corpus.

Parte 3 / 4

Patrones de retrieval

Retrieval con cita: el LLM redacta con referencias.

Retrieval + cita: el LLM redacta con referencias verificables.

Query del usuario:
"*¿Qué comisión de apertura cobra BBVA en hipotecas tipo fijo?*"

[1] Doc 23 (Hipoteca BBVA 2025).pdf *score 0.94*
La hipoteca a tipo fijo incluye una comisión de apertura del 0,5 %...

[2] Doc 47 (Tarifas BBVA general).pdf *score 0.81*
Tarifas vigentes: comisión apertura 0,5 %, sin gastos por amortización...

[3] Doc 12 (FAQ hipotecas).pdf *score 0.62*
Pregunta frecuente 4: ¿qué comisión cobramos al abrir? Respuesta: 0,5 %...

Respuesta del LLM: "BBVA cobra una comisión de apertura del 0,5 % [1]."

Tres patrones que mejoran RAG en producción.

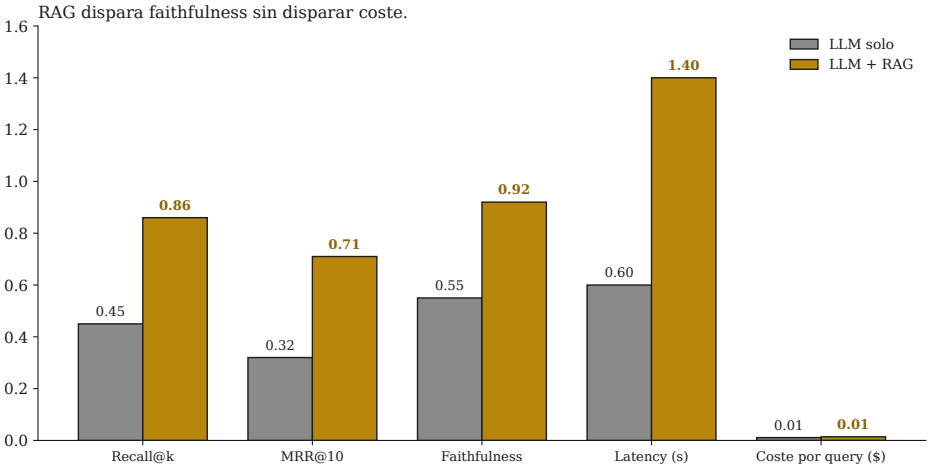
Hybrid search 🔍**.** Combina retrieval semántico (embeddings 📚) con BM25 (keywords). Mejor recall.

Reranking. Tras recuperar 20 chunks, un *cross-encoder* (ej. `bge-reranker-large`) los reordena. Solo los 5 mejores van al LLM 🧠.

Query rewriting. Antes de buscar, el LLM reformula la query del usuario en una más eficaz para retrieval ("descomposición.º "step-back prompting").

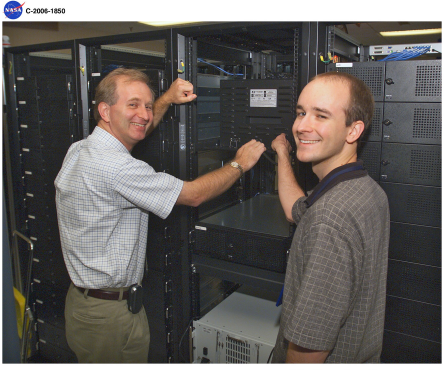
Idea clave. Aplicar estos tres patrones puede subir Recall5 del 0,65 al 0,90 sin cambiar modelo.

Métricas 🚀 que importan en RAG.



Parte 4 / 4

Caso BBVA: dimensiones reales



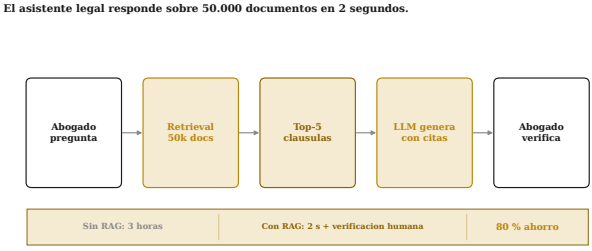
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field

Dimensiones reales aproximadas de un RAG empresarial.

Ejemplo a mano: dimensiones reales aproximadas de un RAG empresarial en BBVA.

Item	Valor
Documentos	120.000 PDFs (circulares + contratos modelo).
Chunks tras chunking	1.450.000 (700 tokens/chunk).
Dim. embedding	1024 (E5-multilingual).
Tamaño Pinecone	11 GB en producción.
Coste ingesta inicial	$890 (re-procesar embeddings).
Coste mensual operativo	$3.200 (Pinecone) + $2.150 (Sonnet).
Faithfulness antes / después	0,57 / 0,93.

El asistente legal responde sobre 50.000 documentos en 2 segundos 📄.



Caso 🔍. Un despacho legal con 50.000 documentos. El abogado pregunta sobre una clausula concreta.

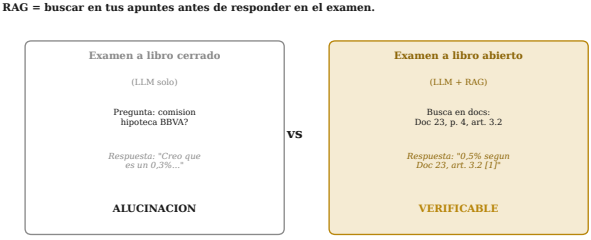
Sin RAG. Busqueda manual: 3 horas entre contratos, circulares, jurisprudencia.

Con RAG. 2 segundos de retrieval + generacion con citas. El abogado verifica la clausula citada.

Ahorro. 80 % del tiempo de busqueda. El abogado se centra en interpretar, no en buscar.

Idea clave. RAG no reemplaza al experto: le devuelve el tiempo para pensar.

RAG = buscar en tus apuntes antes de responder en el examen 📖.



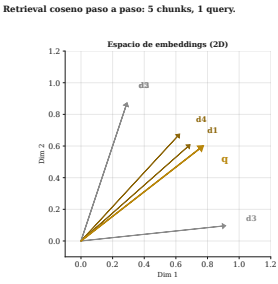
Libro cerrado 🧠. El LLM solo sabe lo que memorizo en entrenamiento. Si no lo vio, inventa (alucina).

Libro abierto. Con RAG, el LLM busca en tus documentos antes de responder. Cita la fuente exacta.

Diferencia clave. El LLM sabe razonar pero no sabe tus datos. RAG le da acceso a tus documentos en cada consulta.

IDEA CLAVE. Examen a libro abierto: la intuicion correcta para explicar RAG a cualquier publico.

Retrieval coseno paso a paso: 5 chunks, 1 query.



Calculo paso a paso:

`q = (0.9, 0.7)`

`||q|| = 1.148`

Doc	Vector	cos(q,d)	Rank
d1	**(0.8,0.8)**	**0.9981**	**#1**
d2	(0.3,0.9)	0.8321	#4
d3	**(0.9,0.1)**	**0.8493**	**#3**
d4	**(0.6,0.7)**	**0.9865**	**#2**
d5	(0.1,0.3)	0.8321	#5

Top-3: d1, d4, d3

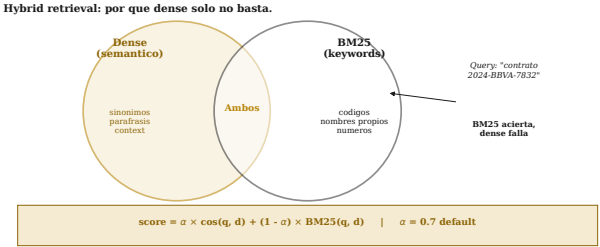
Setup. Query $q = (0{,}9, 0{,}7)$ y 5 documentos en un espacio 2D (proyectado de 1024 dims para visualizar).

Calculo. $\cos(q, d_i) = \frac{q \cdot d_i}{\|q\| \, \|d_i\|}$. Computamos para cada documento, rankeamos, seleccionamos top-3.

Resultado. d_1 y d_4 estan cerca del query (angulo pequeno); d_5 esta lejos (angulo grande). Top-3: d_1, d_4, d_2.

Idea clave. Retrieval es geometria: documentos cercanos al query en el espacio de embeddings.

Hybrid retrieval: por que dense solo no basta 🔍.



Dense 🎯. Captura similitud semantica: sinonimos, parafrasis. Falla con codigos, nombres propios y numeros exactos.

BM25. Busqueda por keywords clasica. Acierta con codigos de producto, IDs, fechas. Falla con reformulaciones.

Hybrid. score $= \alpha \cdot \cos(q, d) + (1-\alpha) \cdot \text{BM25}$. Con $\alpha = 0{,}7$ se cubren ambos casos.

Ejemplo. Query “contrato 2024-BBVA-7832”: dense retrieval pierde el codigo, BM25 lo encuentra.

IDEA CLAVE. En produccion, hybrid retrieval es el default. Dense solo no basta.

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

Concepto 1. RAG separa conocimiento dinámico (documentos 📄) del modelo (estático). Cambio en docs no requiere reentrenar.

Concepto 2. El pipeline canónico tiene seis piezas: ingesta, chunking, embeddings 📚, vector DB, retrieval, LLM con cita.

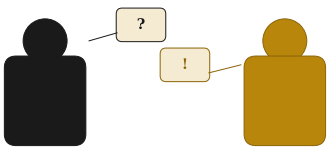
Concepto 3. Para mejorar de verdad, añade hybrid search 🔍, reranking y query rewriting.

Concepto 4. Faithfulness es la métrica clave 🚀 para sectores regulados.

Para la práctica. Montar un RAG mínimo sobre 30-50 PDFs (corporativos o académicos) y medir Recall5.

Próximo día. Día 16 (Tema 6.2): adaptación corporativa (SFT, LoRA, RLHF, DPO).

Pausa para debatir: ¿Pueden las alucinaciones desaparecer del todo con RAG?



La pregunta. RAG cita siempre fuentes. ¿Es suficiente para eliminar alucinaciones?

Posición A. Sí, mientras el chunk recuperado sea relevante.

Posición B. No: el LLM puede malinterpretar o combinar fuentes de forma incoherente.

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

Hasta el Día 16.

Cuándo ajustar el modelo y cuándo no.

Eduardo C. Garrido-Merchán

ecgarrido@comillas.edu