Día 15 · Tema 6 — Ejercicios: RAG y búsqueda semántica

Cómo se usa esta hoja. Hay dos tipos de ejercicio, y sólo dos.

El ejercicio del día. Uno por hoja. Es el obligatorio y es el que entra en el examen, con el cálculo incluido. Basta con hacer ese para seguir el curso con solvencia.

Los ejercicios de ampliación. El resto. Están para enseñar de qué es capaz el método, no para exigir: **no se piden calculados en el examen**. De ellos sólo puede preguntarse la idea, a nivel conceptual.

AMPLIACIÓN *opcional · no se pide calculado en el examen*

1. Chunking de un documento

Un documento tiene un total de 2400 tokens. Se aplica chunking con un tamaño de chunk C y un solapamiento O. El número de chunks se calcula como

$$n = \left\lceil \frac{T - O}{C - O} \right\rceil,$$

donde T es el número total de tokens.

(a) $C = 400$, $O = 0$ (sin solapamiento). ¿Cuántos chunks se generan?
(b) $C = 400$, $O = 100$. ¿Cuántos chunks se generan? ¿Cuántos tokens cubre el último chunk si no está completamente lleno?
(c) $C = 600$, $O = 50$. ¿Cuántos chunks se generan?
(d) Comenta cualitativamente: ¿qué efecto tiene aumentar el solapamiento sobre el número de chunks y sobre la probabilidad de que una idea quede partida entre dos chunks consecutivos?

EL EJERCICIO DEL DÍA *obligatorio · entra en el examen*

2. Recuperación por similitud coseno

Un sistema RAG indexa cinco chunks con embeddings en $\mathbb{R}^4$. El embedding de la consulta del usuario es $\mathbf{q} = (1,\ 0,\ 1,\ 0)$ y los embeddings de los chunks son:

$$\mathbf{d}_1 = (1,1,0,0), \quad \mathbf{d}_2 = (0,0,1,1), \quad \mathbf{d}_3 = (1,0,1,1), \quad \mathbf{d}_4 = (0,1,0,1), \quad \mathbf{d}_5 = (1,0,0,0).$$

(a) Calcula $\cos(\mathbf{q},\ \mathbf{d}_j)$ para cada $j = 1, \ldots, 5$. Recuerda:

$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\,\|\mathbf{v}\|}.$$

(b) Ordena los chunks de mayor a menor similitud.
(c) Si el sistema recupera los top-3, ¿qué chunks se devuelven?

AMPLIACIÓN *opcional · no se pide calculado en el examen*

3. Recuperación híbrida: coseno + BM25

Para los mismos cinco chunks del ejercicio anterior, un motor de búsqueda léxica (BM25) asigna las siguientes puntuaciones (ya normalizadas en $[0, 1]$):

$$\text{BM25}(d_1) = 0{,}8, \quad \text{BM25}(d_2) = 0{,}3, \quad \text{BM25}(d_3) = 0{,}5, \quad \text{BM25}(d_4) = 0{,}9, \quad \text{BM25}(d_5) = 0{,}2.$$

El sistema híbrido combina ambas señales con

$$s_j = \alpha\ \cos(\mathbf{q}, \mathbf{d}_j) + (1 - \alpha)\,\text{BM25}(d_j), \qquad \alpha = 0{,}7.$$

Dado que las similitudes coseno del ejercicio anterior ya están en $[0, 1]$, úsalas directamente (redondea a dos decimales si lo necesitas).

(a) Calcula s_j para cada chunk.
(b) Ordena los chunks de mayor a menor puntuación híbrida.
(c) Compara este ranking con el del ejercicio 2. ¿Qué chunk sube más posiciones gracias a BM25 y cuál baja? ¿Por qué tiene sentido combinar ambas señales?

AMPLIACIÓN *opcional · no se pide calculado en el examen*

© 2026 Eduardo C. Garrido-Merchán · CC BY-NC-SA 4.0

4. Recall@k y MRR

Los chunks relevantes para la consulta anterior son $\{d_2, d_3\}$ (ground truth proporcionada por un anotador humano). Utiliza el ranking por similitud coseno del ejercicio 2.

(a) Calcula Recall@1, Recall@3 y Recall@5. Recuerda:

$$\text{Recall@}k = \frac{|\{\text{relevantes}\} \cap \{\text{top-}k \text{ recuperados}\}|}{|\{\text{relevantes}\}|}.$$

(b) Calcula el MRR (*Mean Reciprocal Rank*). Para una sola consulta, MRR $= 1/r$, donde r es la posición del primer chunk relevante en el ranking.
(c) Si en lugar de usar el ranking coseno usamos el ranking híbrido del ejercicio 3, ¿cambian Recall@3 y MRR? Calcula ambas métricas con el nuevo ranking.

AMPLIACIÓN *opcional · no se pide calculado en el examen*

5. Coste de un sistema RAG por consulta

Un sistema RAG recupera $k = 5$ chunks de 400 tokens cada uno. El prompt del sistema tiene 300 tokens y la consulta del usuario 50 tokens. La respuesta media del LLM es de 400 tokens. Se usa Claude Sonnet con precios de \$3 por millón de tokens de entrada y \$15 por millón de tokens de salida.

(a) Calcula el total de tokens de entrada por consulta.
(b) Calcula el coste por consulta (entrada + salida).
(c) Si el sistema recibe 1 000 consultas al día, ¿cuál es el coste diario? ¿Y el mensual (30 días)?
(d) Si se reduce a $k = 3$ (tras un reranking que descarta 2 chunks), ¿cuánto se ahorra al mes?

AMPLIACIÓN *opcional · no se pide calculado en el examen*

6. Dimensionado de una base de datos vectorial

Un corpus contiene 100 000 documentos con una media de 2 000 tokens cada uno. Se aplica chunking con $C = 400$ y $O = 100$. Los embeddings tienen dimensión $D = 1024$.

(a) Calcula el número total de chunks.
(b) Calcula el almacenamiento necesario para los embeddings en FP32 (4 bytes por float). Expresa el resultado en GB.
(c) Repite el cálculo con cuantización a int8 (1 byte por float).
(d) Calcula el tamaño del texto original (1 token $\approx$ 4 bytes) y compara con el almacenamiento de embeddings. ¿Los embeddings ocupan más o menos que el texto?

AMPLIACIÓN *opcional · no se pide calculado en el examen*

7. Análisis coste-beneficio del reranking

Sin reranking, un sistema RAG tiene Recall@5 = 0,70, lo que se traduce en un 30 % de respuestas erróneas. Cada respuesta errónea genera una escalación al equipo humano con un coste de \$5. Con un cross-encoder que reordena los top-50 candidatos para seleccionar los top-5, el Recall@5 sube a 0,88 y las respuestas erróneas bajan al 12 %. El cross-encoder tiene un coste computacional adicional de \$0,001 por consulta. El sistema recibe 10 000 consultas al mes.

(a) Calcula el coste mensual extra por el cross-encoder.
(b) Calcula el coste mensual de escalaciones *sin* reranking y *con* reranking.
(c) Calcula el ahorro neto mensual (ahorro en escalaciones menos coste del cross-encoder).
(d) ¿A partir de cuántas consultas mensuales el reranking deja de ser rentable? (Pista: plantea la ecuación de punto de equilibrio.)

© 2026 Eduardo C. Garrido-Merchán · CC BY-NC-SA 4.0