Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

Control 3

Integrantes: Adolfo Rojas V.
Profesor: Juan Manuel Barrios
Ayudantes: Andrés Calderón
Martina Navarro
Scarleth Betancurt
Sebastián Sáez

Fecha de realización: 24 de junio de 2025
Fecha de entrega: 24 de junio de 2025
Santiago de Chile

1. Semana 9

1) a. En la tabla 1 se muestran los cálculos para el miniset de palabras

Tabla 1: IDF por palabra con $N = 10,000$

Id	Palabra	Documentos	Inverse Document Frequency
1	artificial	1	$log\left(\frac{10000}{1}\right) = 4$
2	es	1000	$log\left(\frac{10000}{1000}\right) = 1$
3	inteligencia	10	$log\left(\frac{10000}{10}\right) = 3$
4	mi	1000	$log\left(\frac{10000}{1000}\right) = 1$
5	mucha	1000	$log\left(\frac{10000}{1000}\right) = 1$
6	perro	100	$log\left(\frac{10000}{100}\right) = 2$
7	satélite	10	$log\left(\frac{10000}{10}\right) = 3$
8	Suchai	10	$log\left(\frac{10000}{10}\right) = 3$
9	tiene	1000	$log\left(\frac{10000}{1000}\right) = 1$
10	un	1000	$log\left(\frac{10000}{1000}\right) = 1$

b. Por dimensionalidad (y en parte porque tampoco tenemos el vocabulario completo del corpus) trabajaremos en $\mathbb{R}^2$ con *inteligencia* de primera coordenada y *artificial* de segunda

Tabla 2: TF-IDF normalizado

	inteligencia	artificial	vector	norma	$TF - IDF_{norm}$
Q	$1 \cdot 3$	$1 \cdot 4$	$(3, 4)$	$\sqrt{3^2 + 4^2} = 5$	$\left(\frac{3}{5}, \frac{4}{5}\right) = (0.6, 0.8)$
D_1	$0 \cdot 3$	$1 \cdot 4$	$(0, 4)$	4	$(0, 1)$
D_2	$1 \cdot 3$	$0 \cdot 4$	$(3, 0)$	3	$(1, 0)$

c. Dado que nuestros 3 descriptores se encuentran normalizados, la similitud coseno será el producto punto entre estos.

$$\cos(Q, D_1) = 0.6 \cdot 0 + 0.8 \cdot 1 = 0.6 \tag{1a}$$
$$\cos(Q, D_2) = 0.6 \cdot 1 + 0.8 \cdot 0 = 0.8 \tag{1b}$$

d. Utilizando los mismos vectores normalizados tenemos lo siguiente

$$||D_1 - Q||_2 = \sqrt{(0 - 0.6)^2 + (1 - 0.8)^2} \approx 0.632 \tag{2a}$$
$$||D_2 - Q||_2 = \sqrt{(1 - 0.6)^2 + (0 - 0.8)^2} \approx 0.894 \tag{2b}$$

2. Semana 10

1) Lo primero es ver que $|T| = 50,000; |V| = 200$ con $T \subset V$ el conjunto de tokens y V el vocabulario del dataset

   a. En base a esto tenemos que el total de elementos en el unigrama / BoW simple es $|V| = 200$ mientras que para $n = 2$ es $200^2 = 40,000$ por lo que la cantidad de términos totales del bigrama será de $40,200$. En consecuencia la cota superior de la dimensión de vectores TF-IDF es de $40,200$

   b. Calculamos la cantidad de términos con $n = 3$ que es $200^3 = 8,000,000$ por lo que (siendo más formales) el total de términos es $|V_{\leq 3}| := 8,000,000 + 40,200 = 8,040,200$ y se debe cumplir por cota que $\forall \vec{d} \in \text{TF-IDF}_{\leq 3}(\text{V})$, $\dim(\vec{d}) \leq |V_{\leq 3}|$

2) Para facilitar el rellenado de las matrices, cada casilla tiene un par ordenado con la distancia y la operación usada, las abreviaciones de cada operación son $-$: no hacer nada, I : insertar, D : eliminar, RZ : reemplazo entre z-s, RV : reemplazo entre vocales, R : reemplazo para los casos restantes.

   a. salos $\rightarrow$ zorros

		z	o	r	r	o	s
	$(0,-)$	$(2,I)$	$(4,I)$	$(6,I)$	$(8,I)$	$(10,I)$	$(12,I)$
s	$(2,D)$	$(0.3,RZ)$	$(2.3,I)$	$(4.3,I)$	$(6.3,I)$	$(8.3,I)$	$(10,-)$
a	$(4,D)$	$(2.3,D)$	$(0.8,RV)$	$(2.8,I)$	$(4.8,I)$	$(6.8,RV)$	$(8.8,I)$
l	$(6,D)$	$(4.3,D)$	$(2.8,D)$	$(1.8,R)$	$(3.8,R)$	$(5.8,R)$	$(7.8,R)$
o	$(8,D)$	$(6.3,D)$	$(4.3,-)$	$(3.8,R)$	$(2.8,R)$	$(3.8,-)$	$(5.8,I)$
s	$(10,D)$	$(8.3,RZ)$	$(6.3,D)$	$(5.3,R)$	$(4.8,R)$	$(3.8,R)$	$\boxed{(3.8,-)}$

   b. salsas $\rightarrow$ zorros

		z	o	r	r	o	s
	$(0,-)$	$(2,I)$	$(4,I)$	$(6,I)$	$(8,I)$	$(10,I)$	$(12,I)$
s	$(2,D)$	$(0.3,RZ)$	$(2.3,I)$	$(4.3,I)$	$(6.3,I)$	$(8.3,I)$	$(10,-)$
a	$(4,D)$	$(2.3,D)$	$(0.8,RV)$	$(2.8,I)$	$(4.8,I)$	$(6.8,RV)$	$(8.8,I)$
l	$(6,D)$	$(4.3,D)$	$(2.8,D)$	$(1.8,R)$	$(3.8,R)$	$(5.8,R)$	$(7.8,R)$
s	$(8,D)$	$(6.3,RZ)$	$(4.8,D)$	$(3.8,R)$	$(2.8,R)$	$(4.8,R)$	$(5.8,-)$
a	$(10,D)$	$(8.3,D)$	$(6.8,RV)$	$(5.8,R)$	$(4.8,R)$	$(3.3,RV)$	$(5.3,I)$
s	$(12,D)$	$(10.3,RZ)$	$(8.8,D)$	$(7.8,R)$	$(6.8,R)$	$(5.3,D)$	$\boxed{(3.3,-)}$

3) Primero definimos los conceptos importantes: $|\mathcal{D}| = 20,000$. $|V| = 5,000$ con $\mathcal{D}$ el dataset de documentos, V el vocabulario y la matriz $A = [\vec{d_1}, \vec{d_2}, .., \vec{d_{|\mathcal{D}|}}]^T$ con $\vec{d_i}$ el descriptor TF-IDF del i-ésimo documento $D_i \in \mathcal{D}$ por lo que $A \in \mathbb{R}^{20,000 \times 5,000}$

a. De esto tenemos que $\dim(B) = \dim(AA^T) = 20,000^2 = 400,000,000$, donde la matriz resultante contiene el producto punto / similitud entre documentos calculada en base a los términos ($\in V$) comúnes

b. De lo comentado en un principio tenemos que $\dim(C) = \dim(A^TA) = 5,000^2 = 25,000,000$, donde la matriz resultante contiene el producto punto / similitud entre términos a través de todos los documentos ($\in \mathcal{D}$)

c. Simil menos poderoso que embeddings, LSA mejora la efectividad al capturar relaciones semánticas latentes entre términos y documentos, identificando similitudes aunque no compartan términos exactos

d. Simil a PCA, LSA reduce la dimensionalidad del espacio vectorial mediante SVD lo que acelera búsquedas y comparaciones

3. Semana 11

1) a. A continuación el proceso de inserción hecho a mano

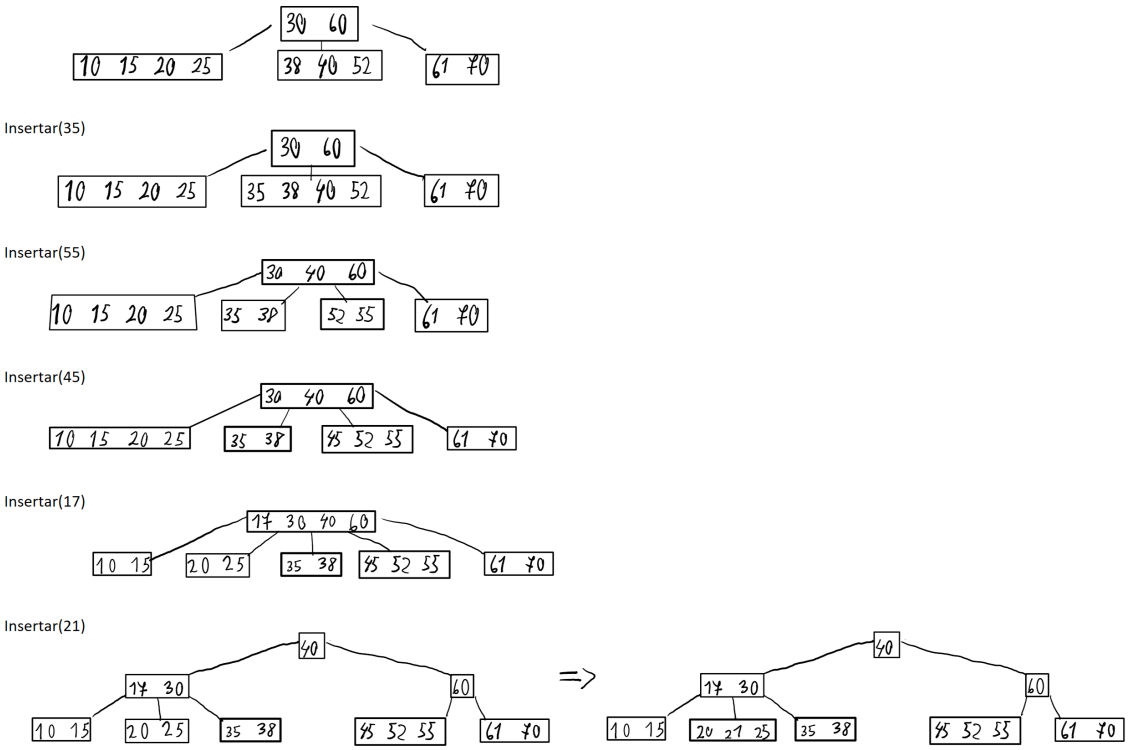

Figura 1: B-tree

2) a. Primero escribimos el resultado de aplicar la hash function $h(x) = \lfloor x/4 \rfloor \mod 6 = (x//4)\,\%6$

En base a esto conseguimos la siguiente tabla

Tabla 3: Resultado de aplicar la función hash

Elemento	$f(x) = \lfloor x/4 \rfloor$	$h(x) = f(x) \mod 6$
10	2	2
24	6	0
37	9	3
40	10	4
46	11	5
56	14	2
72	18	0
76	19	1
84	21	3
92	23	5

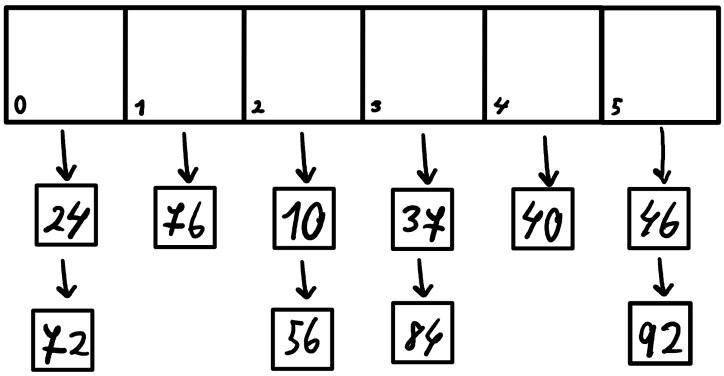

Figura 2: Hash Table resultante

b. Primero calculamos $h(70) = 5, h(75) = 0, h(87) = 3$. Con esto calculamos las distancias de los elementos correspondientes a la cadena compartida, $\min\{|70 - 46|, |70-92|\} = |70-92| = 22$ por lo que el más similar es **92**. Repitiendo el proceso $\min\{|75 - 24|, |75 - 72|\} = |75 - 72| = 3 \implies$ **72**. $\min\{|87 - 10|, |87 - 56|\} = |87 - 56| = 31 \implies$ **56**

4. Semana 12

1) a.
   - Se inicializa heap=(), candidato a NN=null, distancia de corte pruning_dist=∞
   - Se visita nodo raíz y se obtienen regiones R1, R2 y R3
   - Se calculan MINDIST de q a R1, R2, R3 y se agregan a la cola de prioridad
   - heap = (R1, R3, R2)
   - Extraer región de menor MINDIST en la cola: R1
   - Visita R1 y obtiene regiones R13, R12, R11

- Se calculan MINDIST a q a R13, R12, R11 y se agregan a la cola de prioridad (todas son menores que pruning_dist)
- cola de prioridad=(R13, R3, R2, R12, R11)
- Extrae la región de menor MINDIST en la cola: R13
- Visita R13 y obtiene dos puntos
- punto J (compara distancia, es el nuevo candidato a NN, se fija pruning_dist)

b. Por otro lado como tenemos un total de 21 puntos se tiene que Linear Scan debe hacer 21 evaluaciones

2) i. Visitamos la hoja más cercana {G}, luego retornamos el vector G

ii. De i. actualizamos la pruning_dist con G, luego se pasa a la región {B}, como solo hay un vector (B) y este es < pruning_dist, se retorna

iii. De ii. actualizamos pruning_dist con B, pasamos a {L} y retornamos L al ser el más cercano

iv. De iii. actualizamos pruning_dist con L, pasamos a {J, I, H}, comparamos con cada vector de la hoja y retornamos J

v. De iv. actualizamos pruning_dist con J, pasamos a {C, D}, comparamos con cada cada vector de la hoja pero como ninguno mejora pruning_dist retornamos J

vi. Retornamos J (la siguiente región es {A, K}, pero su MINDIST es mayor que pruning_dist)