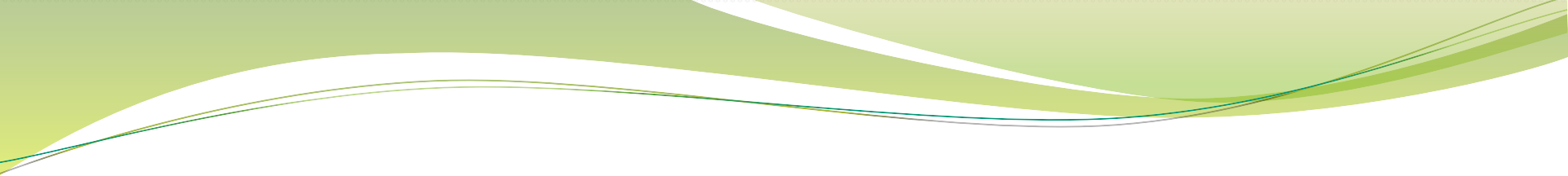


Estructura de Datos y Algoritmos

ITBA 2024-Q2

Lucene

- *Concepto de documento, campos.*
- *Almacenamiento en Lucene: en el índice y fuera del índice*
- *Aplicaciones*
 - *IndexBuilder (creación de los documentos)*
 - *TheSearcher (búsqueda de documentos)*
- *Query:*
 - *API*
 - *QueryBuilder*
- *Formas de separar en tokens*
- *Ranking de documentos*



Empecemos analizando cuál es la fórmula si la consulta está formada por **un solo término**. Es decir:

Query= término

¿Cómo rankear a aquellos documentos que matchean con la consulta?

Query de un término

Dada una colección de N documentos $D = \{DOC1, DOC2, \dots, DOCn\}$ y una query=term, para aquellos documentos que matchean la consulta:

$$\text{Score}(DOC_i, \text{query}) = \text{FormulaLocal}(DOC_i, \text{term}) * \text{FormulaGlobal}(D, \text{term})$$

FormulaGlobal: quiere calcular que tan relevante es esa query en el conjunto de documentos.

Query de un término

FormulaLocal: quiere calcular que tan relevante es esa query respecto a un documento en particular a rankear.

- ¿Es lo mismo si el término aparece pocas veces en el documento que si aparece muchas veces? Intuitivamente si buscan un libro que explique sobre “algoritmos”, es lo mismo si lo menciona 1 vez que si lo menciona 100 veces?
- Por otro lado, es lo mismo si el documento tiene pocos términos que si tiene más términos?

Intuitivamente, ¿es lo mismo si un paper tiene una sola página (ej: 200 términos en ella) y lo menciona 100 veces que si tiene 500 páginas (ej: 100000 términos en ella) y también lo menciona 100 veces? El primero parece ser más específico. O sea, el tamaño del documento es importante.

Query de un término

Teniendo en cuenta lo anterior, es de interés la frecuencia de la aparición del término en el documento normalizado por la longitud del mismo.

En vez de calcular solo el cociente entre ambos, Lucene calcula la raíz cuadrada de dicho cociente porque considera que no es directamente proporcional la relevancia local. Esa fórmula queda:

$$\text{FormulaLocal}(\text{DOC}_i, \text{query}) = \sqrt{\frac{\#freq(\text{term in DOC}_i)}{\#term \text{ existentes en DOC}_i}}$$

Query de un término

FormulaGlobal: quiere calcular que tan relevante es esa query respecto a la colección de documentos existente. Tampoco se quiere linealidad y por eso se aplica un logaritmo.

$$\text{FormulaGlobal}(\text{DOC}, \text{query}) = 1 + \log_e \frac{1 + \# \text{docs en la coleccion}}{1 + \# \text{docs que contienen term}}$$

TP 2-C

Ejer 7



Query de un solo término.

Query de un término

Calcular el ranking de documentos cuando se busca por el término “game” en la colección de documentos formados por el field “content”

La colección es:

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Query de un término

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 2 (c.txt)

game

Docid 3 (d.txt)

Game video,
review game.

$$\text{FormulaGlobal}(\text{DOC}, \text{query}) = 1 + \log_e \frac{1 + \# \text{docs en la coleccion}}{1 + \# \text{docs que contienen term}} = 1 + \log_e \frac{1+4}{1+3} = 1.2231436$$

Value term	Freq en docs	[docid:freqs in docid:[positions in docid]]
game	3	???????
review	1	
store	1	
video	2	

Query de un término

Ahora para aquellos documentos que matchearon la consulta calcular la FormulaLocal

Ahora sí usa la otra parte de la información indexada

<u>Value term</u>	<u>Freq en docs</u>	<u>[docid:freqs in docid:[positions in docid]]</u>
-------------------	---------------------	--

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

Docid 1 (b.txt)

video

Length=1	freq
video	1

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

FormulaLocal(DOC₀,query) =

$$\sqrt{\frac{\#freq(term\ in\ DOC_0)}{\#term\ existentes\ en\ DOC_0}}$$

$$= \sqrt{\frac{1}{2}} = 0.7071067$$

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

$$\begin{aligned}\text{Score}(\text{DOC}_0, \text{query}) &= \\ &\mathbf{0.70710677} \\ &\times \mathbf{1.2231436} \\ &= \mathbf{0.8648931}\end{aligned}$$

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

FormulaLocal(DOC₂,query) =

$$\sqrt{\frac{\#freq(term\ in\ DOC2)}{\#term\ existentes\ en\ DOC2}}$$

$$= \sqrt{\frac{1}{1}} = 1$$

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

$$\text{Score}(\text{DOC}_2, \text{query}) = 1 * 1.2231436 = 1.2231436$$

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

FormulaLocal(DOC₃,query) =

$$\sqrt{\frac{\#freq(term\ in\ DOC3)}{\#term\ existentes\ en\ DOC3}}$$

$$= \sqrt{\frac{2}{4}} = 0.70710677$$

Query de un término

Docid 0 (a.txt)

store,, game

Length=2	freq
store	1
game	1

Docid 2 (c.txt)

game

Length=1	freq
game	1

Docid 3 (d.txt)

Game video,
review game.

Length=4	freq
game	2
video	1
review	1

$$\begin{aligned}\text{Score}(\text{DOC}_3, \text{query}) &= \\ \mathbf{0.70710677} \\ * 1.2231436 &= \mathbf{0.8648931}\end{aligned}$$

Query de un término

Dado que tenemos:

- $\text{Score}(\text{DOC}_0, \text{query}) = 0.8648931$
- $\text{Score}(\text{DOC}_2, \text{query}) = 1.2231436$
- $\text{Score}(\text{DOC}_3, \text{query}) = 0.8648931$

O sea, aparecen ordenados descendentemente por score:

Doc2

Doc0

Doc3



O al revés

Query de un término modificado por fórmula

Si tenemos que el query usa un solo término modificado por FuzzySearch, Range, Prefix, Wildcard : se devuelve como score 1 a los documentos que matchean

Ej: query=ga* sobre los mismos docs

Dado que tenemos:

- $\text{Score}(\text{DOC0}, \text{query}) = 1$
- $\text{Score}(\text{DOC2}, \text{query}) = 1$
- $\text{Score}(\text{DOC3}, \text{query}) = 1$

O sea, aparecen los 3 matching (el score no sirve para ordenar)

Doc0

Doc2

Doc3

Query Multi-término

¿ Cómo calcula el score si la consulta incluye más de un término?

Para aquellos documentos que matchearon la consulta les aplica la siguiente fórmula:

$$\text{Score}(\text{DOC}_i, \text{query}) = \sum_{\text{term in query y no tiene NOT}} \text{FormulaLocal}(\text{DOC}_i, \text{term}) * \text{FormulaGlobal}(\text{D}, \text{term})$$

O sea, se hace la sumatoria del cálculos parciales de los scores de cada término participante en la query, siempre que no esté modificado por NOT

TP 2-C

Ejer 8



Query multi término.

Query Multi-término

Calcular el ranking de documentos cuando se busca por el término “game AND NOT store” en la colección de documentos formados por el field “content”. La colección es la misma:

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Rta

¿Cuáles son los documentos que matchean la query?

~~Docid 0 (a.txt)~~

~~store,, game~~

~~Docid 1 (b.txt)~~

~~video~~

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

El score debido al término “game” ya lo hemos calculado.

El score debido al término “store” no aplica a la fórmula de score porque está afectado por NOT. Obtendremos para esos 2 documentos los valores de score anteriores.

- $\text{Score}(\text{DOC2}, \text{query}) = 1.2231436$
- $\text{Score}(\text{DOC3}, \text{query}) = 0.8648931$

Y en el resultado primero aparece Doc2 y luego Doc3

Query Multi-término

Calcular el ranking de documentos cuando se busca por el término “game AND store” en la colección de documentos formados por el field “content”. La colección es la misma:

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Query Multi-término

Rta

¿Cuáles son los documentos que matchean la query?

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Query Multi-término

$$\text{FormulaGlobal}(\text{DOC}, "store") = 1 + \log_e \frac{1 + \# \text{docs en la coleccion}}{1 + \# \text{docs que contienen "store"}}$$

$$= 1 + \log_e \frac{1+4}{1+1} = 1.9162907$$

Value term	Freq en docs	[docid:freqs in docid:[positions in docid]]
game	3	??????
review	1	
store	1	
video	2	

Query Multi-término

Docid 0 (a.txt)

store game

Length=2	freq
store	1
game	1

FormulaLocal(DOC₀, "store")

$$= \sqrt{\frac{\#freq("store" \text{ in } DOC_0)}{\#terms \text{ existentes en } DOC_0}}$$

$$= \sqrt{\frac{1}{2}} = 0.7071067$$

Query Multi-término

Docid 0 (a.txt)

store game

Length=2	freq
store	1
game	1

$$\begin{aligned}\text{Score}(\text{DOC}_0, \text{query}) &= \\ &\mathbf{0.70710677} \\ &\times \mathbf{1.9162907} \\ &= \mathbf{1.3550219}\end{aligned}$$

Finalmente

$$\begin{aligned}\text{Score}(\text{DOC}_0, \text{"game AND store"}) &= \mathbf{0.8648931} + \mathbf{1.3550219} \\ &= \mathbf{2,219915}\end{aligned}$$

Query Multi-término

Calcular el ranking de documentos cuando se busca por el término “game OR store” en la colección de documentos formados por el field “content”

La colección es la misma:

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Query Multi-término

Rta

¿Cuáles son los documentos que matchean la query?

Docid 0 (a.txt)

store,, game

Docid 1 (b.txt)

video

Docid 3 (d.txt)

Game video,
review game.

Docid 2 (c.txt)

game

Query Multi-término

El calculo de $\text{FormulaGlobal}(\text{DOC}, \text{"store"}) = 1.9162907$ YA LO TENIAMOS

Falta calcular FormulaLocal para los nuevos docs (para doc0 ya lo teníamos)

Como en doc3 y doc2 no estaba "store",

$\text{FormulaGlobal}(\text{DOC3}, \text{"store"}) = 0$

$\text{FormulaGlobal}(\text{DOC2}, \text{"store"}) = 0$

Pero la parte de "game" si les da score (el que calculamos previamente)

Query Multi-término

Finalmente

$$\text{Score}(\text{DOC0}, \text{"game OR store"}) = 0.8648931 + 1.3550219 = 2.219915$$

$$\text{Score}(\text{DOC2}, \text{"game OR store"}) = 1.2231436 + 0 = 1.2231436$$

$$\text{Score}(\text{DOC3}, \text{"game OR store"}) = 0.8648931 + 0 = 0.8648931$$

Y Rankean:

Doc0

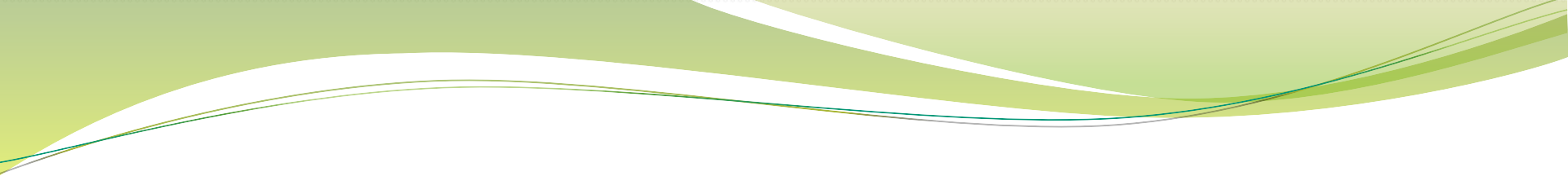
Doc2

Doc3

Consideraciones

Lucene es un excelente implementación para Fulltext Retrieval.

Permite que lo extendamos. Ej: podemos hacer una Analyzer que maneje Metaphone para ofrecer búsqueda fonética.



Lucene no ofrece escalabilidad. A medida que el conjunto de documentos crece o los clientes que realizan consultas crece, eso puede ser un problema.

Si se precisa un backend que permita escalabilidad, debemos usar **Solr** o **Elasticsearch**. Ambos frameworks están contruidos sobre Lucene pero permiten escalar ya que coordinan varias instancias de Lucene, las cuales pueden correr en diferentes computadoras (en un cluster de computadoras).

Lucene



Index
lucene:
doc1, doc2,
doc3, doc4,
doc5, doc6

Docs:
Doc1, doc2,
doc3, do4,
doc5, doc6

Solr / Elasticsearch

