Lezione 7 - Indicizzazione e recupero dei documenti di testo

https://www.youtube.com/watch?v=EYv_3U9d5ps

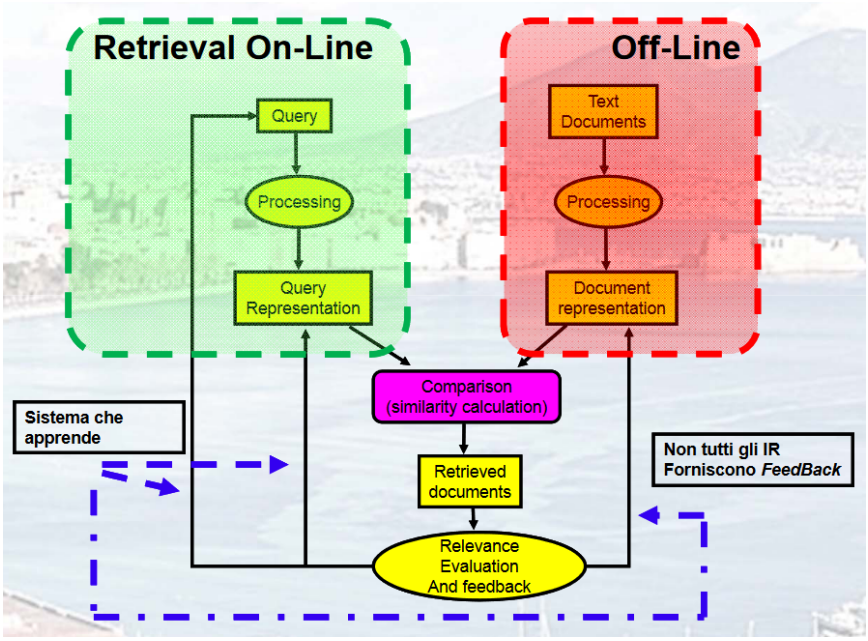
Differenze tra IR e DBMS

DMBS

- Struttura omogenea di record
- Ogni record ha dei componenti: attributi
- Un record è definito completamente e univocamente
- Retrieval con match esatto

IR

- Records non strutturati
- Attributi non prefissati
- Indicizzazione del documento:
  - Keywords
  - Descrittori
  - Indici
- Dipendenza dal modello di rappresentazione dei documenti
- Retrieval con match approssimato o parziale



Processo base del document retrieval

Indicizzazione automatica e modello booleano

Il modello ri retrieval booleano sono utilizzati su funzioni semplici (es: comandi di shell unix)

```
/[^a-z]a/
grep root /etc/passwd
```

Struttura File

Bisogna cercare un'adeguata rappresentazione della conoscenza:

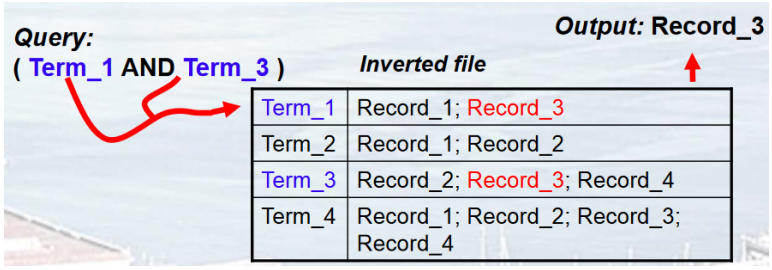
- Flat file → Classici file ASCII, sui quali è possibile eseguire ricerche lineari (grep, awk)
- Inverted files → Estensione dei flat
- Signature files → Genera impronte e confronta quelle della query con quelle del documento
- Alberi
- Grafi

Inverted Files

Tipo di file che contiene un insieme di righe di testo. Ogni riga contiene:

- Il termine da ricercare
- Una sequenza di puntatori a documenti e record che contengono quel termine

“Inverted” indica quindi l’inversione del verso di ricerca: prima la chiave poi il documento che la contiene



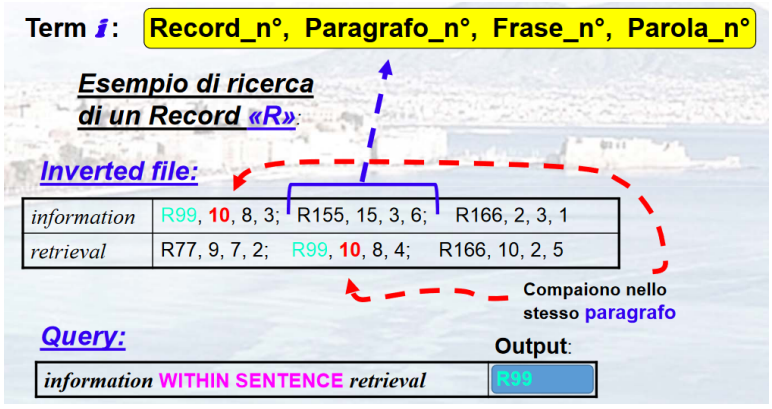
Esempio di ricerca con file invertito

Il processo di ricerca è più efficiente rispetto al flat file: non si analizzano i documenti interi ma solo l’inverted file da cui si ricavano i collegamenti ai documenti che contengono la chiave

Possono essere definite anche delle operazioni estese:

- Within sentence → Due termini sono presenti nello stesso paragrafo
- Adjacent → Due termini confinanti nel record recuperato

Questi sono detti “operatori di prossimità”



Struttura generale dell’inverted file esteso

Indicizzazione Automatica

Il processo di indicizzazione del file prevede diverse fasi di filtraggio:

1. Stop Words → Si escludono elementi insignificanti per la ricerca (es: articoli)

2. Stemming → Si considera solo il termine comune delle parole analoghe (es: pesca-pescatore)
3. Thesaurus → Possibilità di sostituire diversi termini simili che compaiono nel testo con un unico termine (es: sostituire “lavare”, “pulire” e “detergere” con “lavare”)
4. Weighting → I termini che compaiono nel testo hanno diversa importanza, che può essere ricavata valutando le loro frequenze di occorrenza

Calcolo dei pesi

Dato W_{ij} il peso del termine j nel documento i, posso calcolare il peso del termine con:

$$W_{ij} = tf_{ij} * \log(\frac{N}{df_j})$$

Dove:

- tf_{ji} è la frequenza del termine j nel documento
- N è il numero totale dei documenti nel DB
- df_j è il numero dei documenti del DB che contengono il termine j

Se il termine è presenti in tutti i documenti del DB allora $df_j = N$, da cui deriva che $W_i = 0$. Ciò mi dice che il termine non ha un alto potere discriminante.

Al contrario, se il termine è presente in pochi documenti, il suo potere discriminante salirà

Modello Spazio Vettoriale

Si effettua il retrieval utilizzando il prodotto scalare:

$$|a||b|\cos(\theta) \qquad\qquad a * b = \sum_{i=1}^{n} a_i b_i$$

> Due vettori sono più simili quanto più l’angolo fra di loro è stretto

Calcolo della similarità normalizzata:

$$S(D_i, Q_i) = \frac{\sum_{k=1}^{N}(T_{ik} * Q_{jk})}{\sqrt{\sum_{k=1}^{N}(T_{ik}^2) * \sum_{k=1}^{N}(Q_{jk}^2)}}$$

In questa formula:

- il valore $S(D_i, Q_i)$ rappresenta proprio il $\cos(\theta)$ nella formula del prodotto scalare
- Il numeratore rappresenta il prodotto scalare tra le componenti dei due vettori
- Il denominatore è il calcolo del modulo di ogni componente nel prodotto scalare

Attraverso questa formula dimostro l'uguaglianza delle due formule del prodotto scalare

> Quando il coseno tende a 0 la similitudine è alta, se tende a 1 il contrario

Esempio di modello spazio vettoriale

Sono dati 4 documenti rappresentati con i vettori:

$$D_1 = [0.2, 0.1, 0.4, 0.5]$$
$$D_2 = [0.5, 0.6, 0.3, 0]$$
$$D_3 = [0.4, 0.5, 0.8, 0.3]$$
$$D_4 = [0.1, 0, 0.7, 0.8]$$

e sia assegnata una Query rappresentata dal vettore:

$$Q = [0.5, 0.5, 0, 0]$$

Posso trovare, usando le formule, le misure di similarità:

$$S(D_1, Q) = 0.31$$
$$S(D_2, Q) = 0.93$$
$$S(D_3, Q) = 0.66$$
$$S(D_4, Q) = 0.07$$

Il sistema presenterà i documenti: $D_2 \rightarrow D_3 \rightarrow D_1 \rightarrow D_4$

Relevance Feedback

Il motore cerca di sfruttare il feedback che l'utente gli fornisce con le sue query, e i termini ritenuti irrilevanti vengono cancellati dalla query

Si applica la Formula di Rocchio:

$$\vec{Q_m} = (a * \vec{Q_o}) + (b * \frac{1}{|D_r|} * \sum_{\vec{D_j} \in D_r} \vec{D_j}) - (c * \frac{1}{|D_{nr}|} * \sum_{\vec{D_k} \in D_{nr}} \vec{D_k})$$

Variable	Value
$\overrightarrow{Q_m}$	Modified Query Vector
$\overrightarrow{Q_o}$	Original Query Vector
$\overrightarrow{D_j}$	Related Document Vector
$\overrightarrow{D_k}$	Non-Related Document Vector

Variable	Value
a	Original Query Weight
b	Related Documents Weight
c	Non-Related Documents Weight
D_r	Set of Related Documents
D_{nr}	Set of Non-Related Documents

Valori utilizzati nella formula di Rocchio

Quello che fa il Relevance Feedback non è modificare la query, ma modificare la serie dei documenti che devono essere presentati

Generazione del Cluster

Similarità per coppie

- Ogni documento è rappresentato come un vettore
- Si calcola la similitudine per ogni coppia di documenti e si popola una matrice delle distanze
- Si prende la coppia con la distanza minore e si elimina dalla matrice
- Si ripete questo procedimento finché la matrice non sarà composta da un solo elemento

Clustering Euristico

- Il primo documento sarà il primo cluster
- Ogni documento successivo viene confrontato con i cluster generati e viene collocato nel cluster più vicino
- Si ripete fino all'esaurimento dei documenti