

Social MINING

IIº SEM

2021



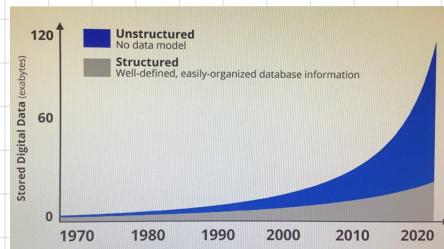
11/03

INFORMATION RETRIEVAL (IR)

IR - TASK

STRUCTURED vs
UNSTRUCTURED DATA

- Trovarci materiali (docs, video, images...)
- Spesso i dati sono **UNSTRUCTURED**, addirittura nelle grandi compagnie
 - ↳ **SEMI-STRUCTURED** sono ad esempio le slides con **FIELDS** come **TITLO** e **CONTENUTO**.
- **STRUCTURED**: Relational DBs, XML, ...



IR riguarda solo i 9
RETRIEVAL? → NO

- **CLOUDMINING** cluster di documenti
- **CATEGORIZATION** dato un insieme di topic assegnargli un documento
- **INFORMATION EXTRACTION**
- **QUESTION ANSWERING** Rispondere a domande come: facts, How, Why
- **OPINION MINING** analizzare il sentimento dietro i testi

Terminology

SEARCHING: ricerca di una **specifico** informazione

BROWSING: **UNSTRUCTURED EXPLORATION**

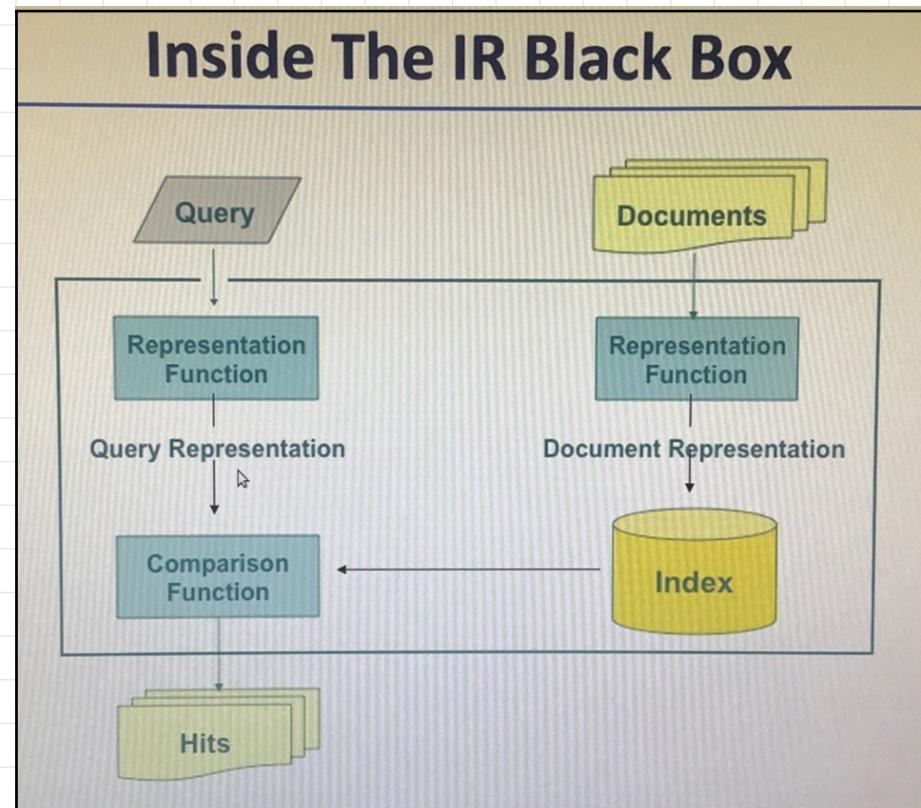
CRAWLING: muoversi tra **HYPYERLINKS**

SCRAPING: estrarre contenuto da pagine

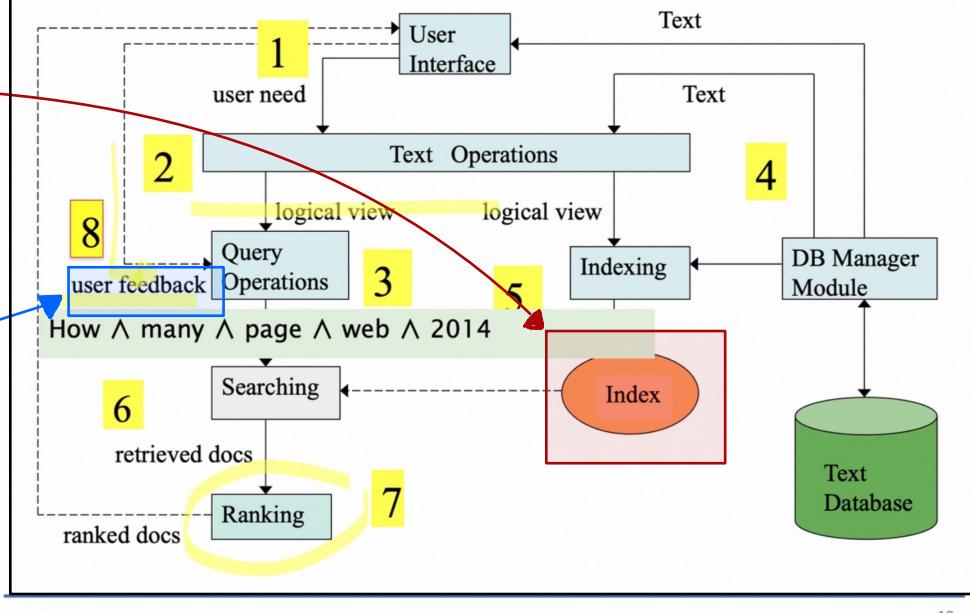
QUERY: **Stringa**

FULL-TEXT SEARCHING: comporre la query con ogni parola del testo

FIELDED SEARCH



Workflow



L'INDEX è ciò che vogliamo creare al fine di indirizzare la query

Se l'utente dica un risultato, gli abbiamo dato ciò che cercava.

es. Può dipendere da FEEDBACK (oppure da PARTE RANK)

• Ottenuti i risultati come li presentiamo all'utente?

SORTING

RANKING by similarity tra query e documenti

RANKING by importance

DOCUMENT REPRESENTATION

. Dato un documento non strutturato \Rightarrow strutturato

(1) BAG OF WORDS MODEL : Sia un array dei Token

LDVARIANTI

HOW TO WEIGHT A WORD

\hookrightarrow es. BOOLEAN : Data una lista, assegna 1 se la Parola è nel documento

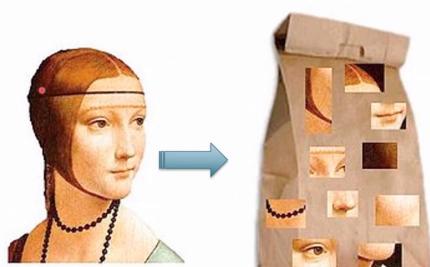
WHAT IS A WORD

\hookrightarrow SINGLE WORD

DOUBLE " : Nome e cognome

INCONDIZIONE : {go, gone, going...}, o
{polite, courtesy, dormitory, casa}

\hookrightarrow usato anche con immagini



DOCUMENT PARSENG

- composta scanning e trasformazione in bag of words
- ci servono 3 cose:
 - lingua (esiste MULTILINGUALITY)
 - tipo di file
 - character set
- {
 - è un singolo file?
 - è una singola pagina web o un sito?

UNIT DOCUMENT

Cos'è un
unità di
documento?

TOKENIZATION

ISSUES

- Un TOKEN è una sequenza di caratteri

- Come scegli i token?

- ↳ Prendo <Nome> - <sogno> insieme?
- ↳ Prendo punteggiatura?
- ↳ Prendo STOP WORDS?
- ↳ i numeri (DATE, NUM. DI TELEFONO, ...)
- ↳ Lingue (arabo, cinese, ...)

APPROCCI

STOP WORDS
• AND, A, TO,
↳ tempo stopgo?

NORMALIZATION

ACRONIMO

- U.S.A. => USA

SINONIMO

- CAR => AUTOMOBILE

TYPO

- GOOLGE => GOOGLE

CASE-FOLDING

- tutto in lowercase

LEMMATIZATION

- Riduzione a un LEMMA

es. {are, is, am} → be

{car, cars, car's} → car

STEMMING

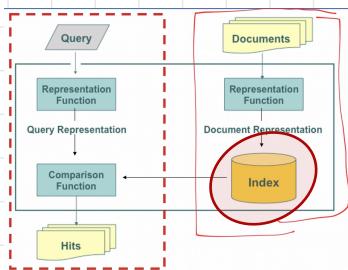
- Riduzione alla RADICE comune

for example compressed
and compression are both
accepted as equivalent to
compress.

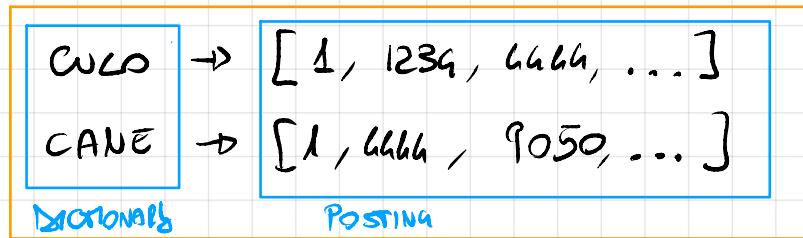
for example compress and
compress are both accepted
as equivalent to compress

17/03

INVERTED INDEX



- L'index è una lista che mappa ogni termine agli IDs dei documenti che lo contengono (POSTING LIST). È detto INVERTED perché non mappa DOC → TERMS MA TERM. → Docs.

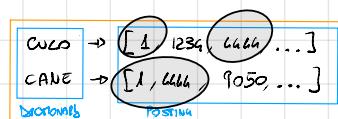


- Inoltre nel DICTIONARY non teniamo le occorrenze del termine totale, ma solo il numero di documenti in cui appare

INDEX SEARCH

Query

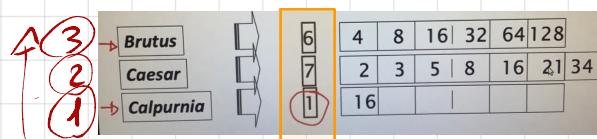
- query: "CUCO AND CANE"
→ Merghiamo i Postings di entrambi e riportiamo quindi i Docs {1, 6666}



OPTIMIZZAZIONE

- Merghiamo a partire da quello con POSTING più corto.

Lo ecco perché abbiamo tenuto i **num. di docs** in cui appare, così possiamo sapere a priori l'ordine di MERGING



PHRASE query

- query: "RED BRICK HOUSE" → non va bene, il modo di prima, perché le parole sono legate

BI-WORD INDEXIS

- Rappresentiamo i due dizionari con dopp. termini
- Ad esempio dalla query: "STANFORD UNIVERSITY PAO ALTO" esco:

"STANFORD UNIVERSITY" AND "UNIVERSITY PAO" AND "PAO ALTO"

- Ovviamente la grandezza dell'INDEX cresce di bretto

- Per ogni termine stiamo sia i doc sia la posizione del termine nel documento, e teniamo anche la doc.freq.

```
<term, number of docs containing term;
doc1: position1, position2 ... ;
doc2: position1, position2 ... ;
etc.>
```

to, 993427: *

- 1, 6: (7, 18, 33, 72, 86, 231);
- 2, 5: (1, 17, 74, 222, 255);
- 4, 5: (8, 16, 190, 429, 433);
- 5, 2: (363, 367);
- 7, 3: (13, 23, 191); ...)

be, 178239:

- 1, 2: (17, 25);
- 4, 5: (17, 191, 291, 430, 434);
- 5, 3: (14, 19, 101); ...)

Ad esempio per la query: "To BE or NOT To BE"

- Extract inverted index entries for each distinct term: **to, be, or, not.**
- Merge their **doc:position** lists to enumerate all positions with "to be or not to be".

• **to:**

- 2:1,17,74,222,551;
- 4:8,16,190,429,433;
- 7:13,23,191; ...

• **be:**

- 1:17,19;
- 4:17,191,291,430,434; 5:14,19,101; ...

① MELCO I DOCUMENTI
IN CW APPARE

② CONTROLLA SE LE
POSIZIONI IN UNO DOC.
SONO CONSECUTIVE

RELAXATION

- Possiamo rilassare la tecnica riferendoci a una certa distanza nel documento

- For example: *employment /4 place*
Find all documents that contain EMPLOYMENT and PLACE within 4 words of each other.
- "*Employment agencies that place healthcare workers are seeing growth*" is a hit.
- "*Employment agencies that have learned to adapt now place healthcare workers*" is **not a hit.**

RANKING

BOOLEAN MODEL

- Query e Docs rappresentati come espressioni booleane



$$\begin{aligned} q = a \wedge (b \vee (\neg c)) &= \\ (a \wedge b \wedge c) \vee (a \wedge b \wedge (\neg c)) \vee (a \wedge (\neg b) \wedge (\neg c)) &\quad (\text{DNF form}) \\ \square \vee (q_{\text{dnf}}) &= (1,1,1) \quad (1,1,0) \quad (1,0,0) \\ \gg \text{Disjunctive Normal Form} \end{aligned}$$

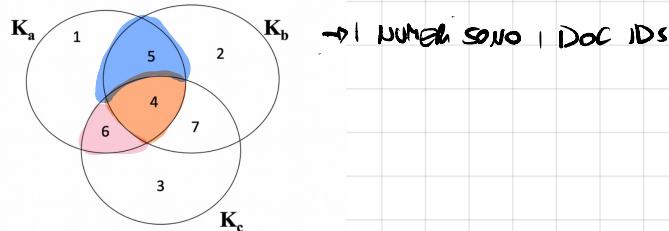
» Ex: $(apple, computer, red) \vee (apple, computer) \vee (apple)$

$$\square \vee (q_{\text{cc}}) = (1,1,0)$$

» Conjunctive Component

c'è un MATH perché i
documenti ci interessano

- Similar/Matching documents
- $md_1 = [apple\ apple\ blue\ day]$ $\Rightarrow (1,0,0)$
- $md_2 = [apple\ computer\ red]$ $\Rightarrow (1,1,1)$



$$\begin{aligned} q = k_a \wedge (k_b \vee k_c) \\ (1 \wedge 1 \wedge 1) \vee (1 \wedge 1 \wedge 0) \vee (1 \wedge 0 \wedge 1) \end{aligned}$$

Which one?

- Il problema è che tutti i documenti sono allo stesso livello, e magari a una query possono essere radati TROPPI / POCO risultati. Perché un documento o match o non lo fa quindi questi query generano risultati.
- Però se cercassimo un metodo di scoring delle query, TROPPI risultati non sarebbero un problema, perché l'utente guarda i primi risultati.

OBS

VECTOR WEIGHTED MODEL

- Abbiamo un modello **BAG OF WORDS VECTOR N-DIMENSIONALE** dove $|V|$ è la dimensione del vocabolario, abbiamo inoltre un **WEIGHTING SCHEME** $w_{t,d}$ che esprime la frequenza della TERM t nel DOCUMENTO d .

Come siamo un valore w ?

TERM FREQ.

CUMULATIVE
MODEL

- Frequenza $tf_{t,d}$ del termine t nel documento d

- Ci serve una sommatoria tra query e documento, sommiamo l'occorrenza di t tra i termini che q e d hanno in comune

$$\text{sim}(q,d) = \sum_{t \in q \cap d} tf_{t,d}$$

- Dobbiamo normalizzare i TERM FREQ.

$$tf'_{t,d} = tf_{t,d} / \sum_{i \in d} (tf_i)$$

$$\text{sim}(q,d) = \sum_{t \in q \cap d} (tf'_{t,d} / \sum_{i \in d} (tf_i))$$

- Alternativa alla normalizzazione

$$w_{t,d} = \sum_{i=0}^{\infty} (1 + \log tf_{t,d}) \quad \text{if } tf_{t,d} > 0 \\ \text{else } 0$$

$$\text{sim}(q,d) = \sum_{t \in q \cap d} (1 + \log tf_{t,d})$$

es
 $0 \rightarrow 0$
 $1 \rightarrow 1$
 $2 \rightarrow 1.3$
 $10 \rightarrow 2$
 $1000 \rightarrow 4$



LOG FREQUENCY

È abbastanza?

- Non ci stiamo \rightarrow , termini rari daranno più informazioni di quei frequenti

INVERSE DOCUMENT FREQUENCY

NOTA: C'è anche la collection
fissa, cioè il numero di
occorrenze in tutta la
collectione (non è il numero
dei documenti in cui appare)

- f_d è il numero di documenti che contiene t in una collezione di N documenti (DOCUMENT FREQUENCY)

- L' INVERSE DOC. FREQUENCY e :

$$\text{idg}_t = \log_{10} N/dg_t$$

Quindi si prevede che li calcolano alla fine?

- ⇒ MIKIANO | DUE
KONI

$$w_{+,d} = (1 + \log t_{f+,d}) \cdot \log N/f$$

CUMULATIVE APPROACH CON LGN TREE.

INVERSE TONE FREQUENCY

* Più alta è la più i termini i comuni

*Più alte è N/di più tardi testimonie e Rep.

* Più alto è w_t , più i termini i ER sono

- So we have a $|V|$ -dimensional vector space, one dimension for each term.
 - Terms are axes of the space
 - Documents are points or vectors in this space.
 - The coordinate of a vector d_j on dimension i is the tf-idf weight of word i in document j .
 - **Very high-dimensional:** hundreds of thousands of dimensions when you apply this to a web search engine
 - It is a very sparse vector - most entries are zero (will see later)

