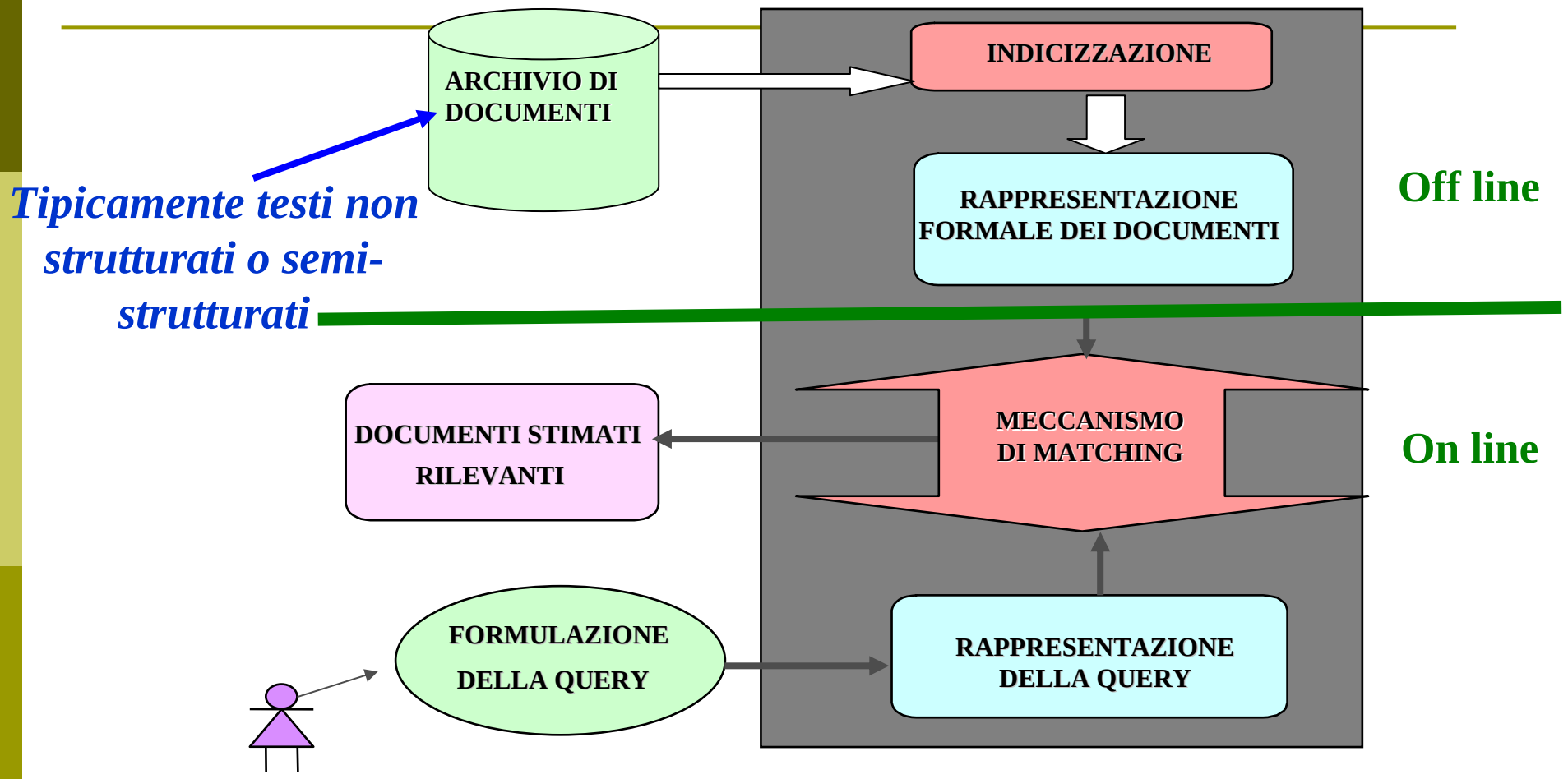
Modelli di Information Retrieval: I modelli base

Gabriella Pasi

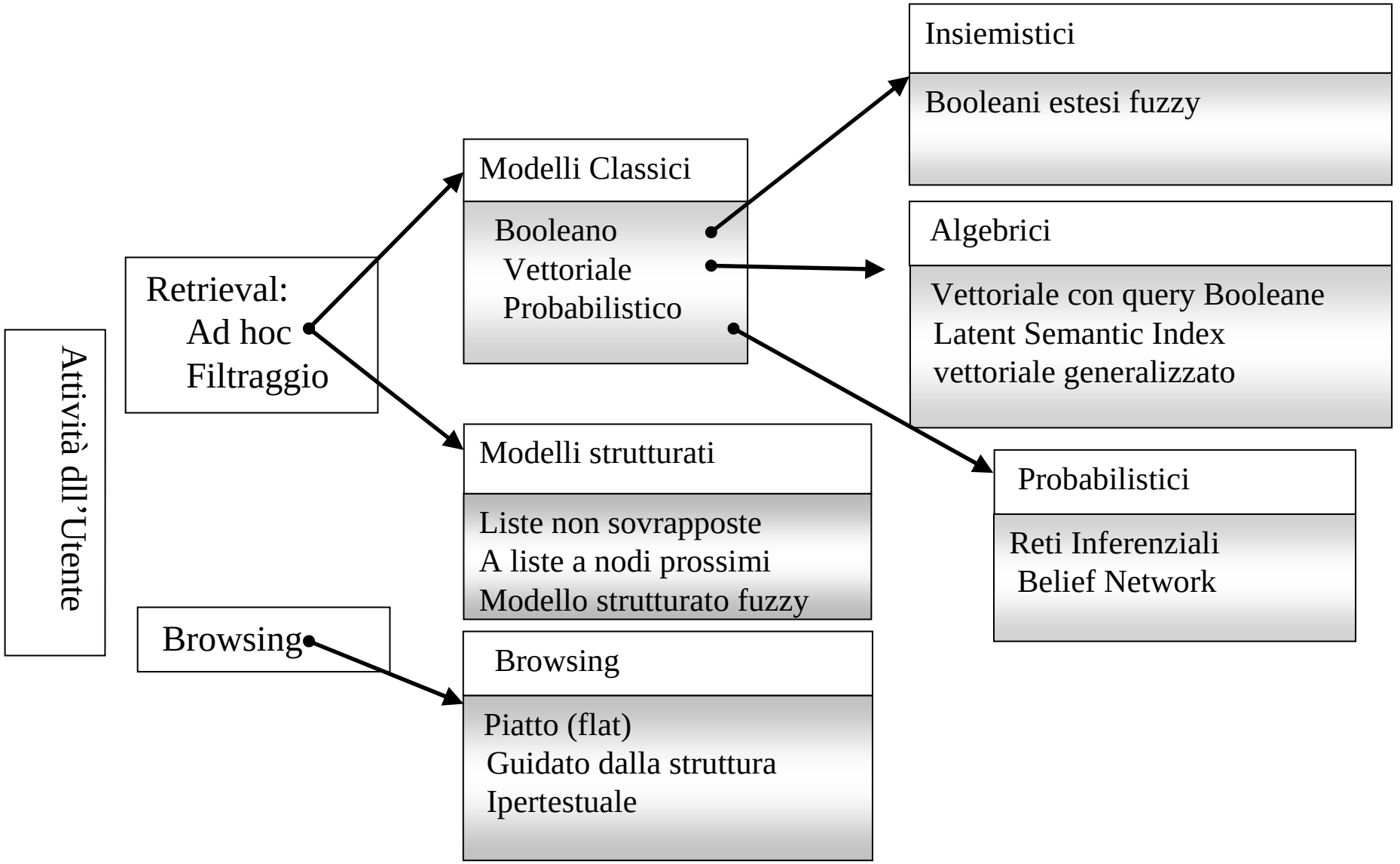
[1] Università degli Studi di Milano Bicocca
Via Bicocca degli Arcimboldi 8
e-mail: pasi@disco.unimib.it

Struttura base di un IRS



Un IRS è basato su un modello matematico

Modelli di IR



Modello Booleano di IR
Concetti di Base

- E' basato sulla teoria degli insiemi
- L''***importanza (significatività)*** dei termini indice è rappresentata da **pesi binari**: $w_{kj} \in \{0,1\}$ peso associato alla coppia $(t_k d_j)$, *dove* t_k è un termine indice e
- d_j - un documento è rappresentato da un insieme di termini $R(d_j)$:

$$R(d_j) := \{t_k \mid w_{kj} = 1 \ \ e \ k=1,..n\};$$

- Una query è formalmente definita come ***espressione Booleana su termini*** (uso degli operatori Booleani AND, OR e NOT) e definisce in modo preciso l'insieme di documenti da selezionare
- Il meccanismo di matching applica ***operazioni insiemistiche***
- La rilevanza à modellata come ***proprietà binaria*** dei documenti (Retrieval Status Value 0 oppure 1)

Linguaggio Booleano

query Booleana : due o più termini di ricerca connessi da operatori Booleani

and *or*

I termini possono essere negati mediante operatore *not*

Esempi:

abacus *and* actor

abacus *or* actor

(abacus *and* actor) *or* (abacus *and* atoll)

not actor

Matching nel modello Booleano

Rappresentazione dei Documenti: insiemi di termini

$R_{d1}= \{t_1, t_2, t_3\}$
$R_{d2}= \{t_1, t_4, t_5\}$
$R_{d3}= \{t_2, t_5\}$

$\Rightarrow$inversione della rappresentazione

Posting di un termine: insiemi di documenti circa un dato concetto

$R_{t1}=\{d_1, d_2\}$
$R_{t2}=\{d_1, d_3\}$
$R_{t3}=\{d_1\}$

Confronto esatto: valuta le operazioni insiemistiche

$q = t_1$ $\quad R_{t1}=\{d_1, d_2\}$

$q = t_1$ AND t_2 $\quad R_{t1} \cap R_{t2} = d_1$

$q = t_1$ OR t_2 $\quad R_{t1} \cup R_{t2} = \{d_1, d_2, d_3\}$

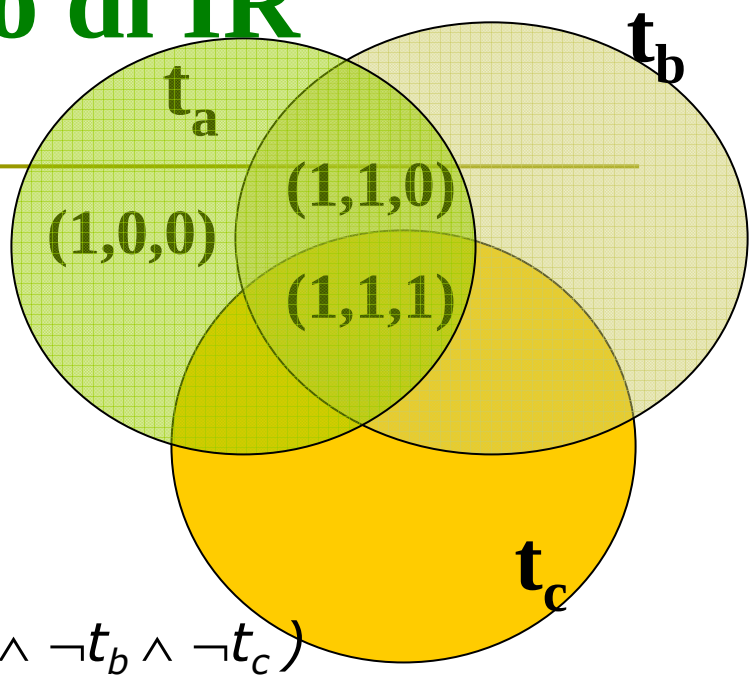
$q =$ NOT t_1 $\quad \neg R_{t1} = d_3$

Modello Booleano di IR

- Consideriamo la query:
  - $q = t_a \wedge (t_b \vee \neg t_c) =$
  - *Ogni query Booleana può essere riscritta in forma normale disgiuntiva (Disjunctive Normal Form).*
  - $q_{dnf} = (t_a \wedge t_b \wedge t_c) \vee (t_a \wedge t_b \wedge \neg t_c) \vee (t_a \wedge \neg t_b \wedge \neg t_c)$
  - q_{cc}= è un disgiunto
  - *Ogni disgiunto rappresenta un insieme di documenti ideali*
  - *La query è soddisfatta da un documento se questi è uno dei documenti ideali di un disgiunto*

Modello Booleano di IR

- $q = t_a \wedge (t_b \vee \neg t_c)$



$q_{dnf} = (t_a \wedge t_b \wedge t_c) \vee (t_a \wedge t_b \wedge \neg t_c) \vee (t_a \wedge \neg t_b \wedge \neg t_c)$

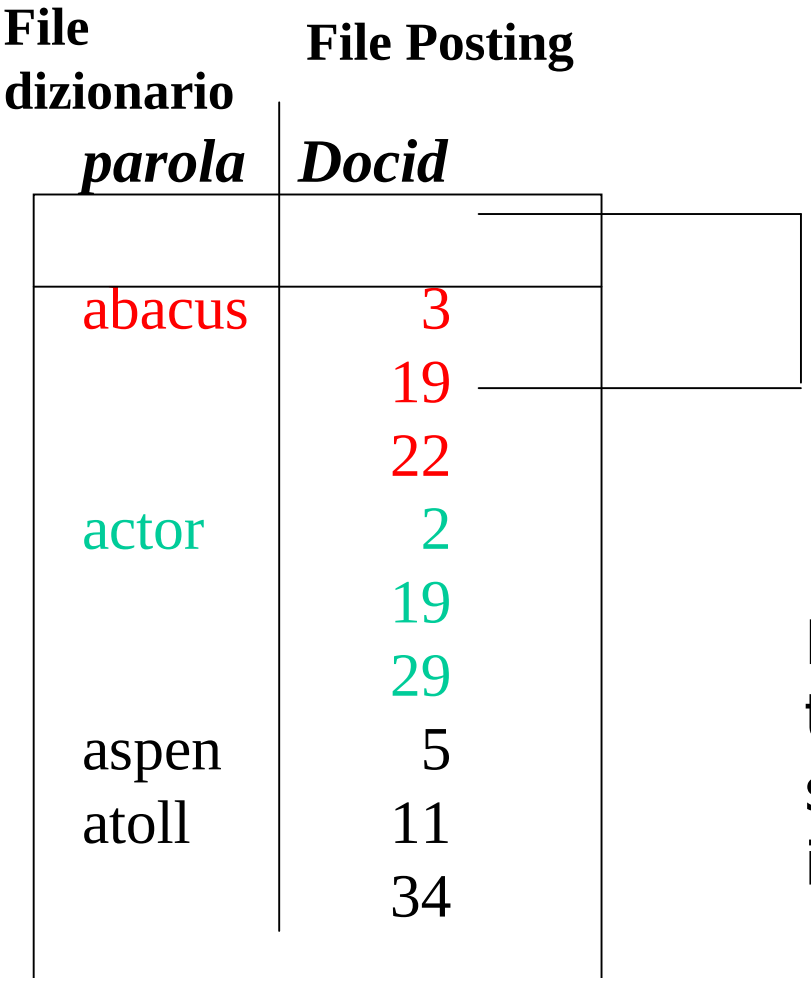
$q_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$

sim(q,d_j) = 1 *se* $\exists\, q_{dg} \mid (q_{dg} \in q_{dnf}) \wedge (\forall t_i \in q_{dg} \rightarrow w_{ij} = 1 \wedge \forall \neg t_i \in q_{dg} \rightarrow w_{ij} = 0)$

= 0 *altrimenti*

Valutazione di query Booleane

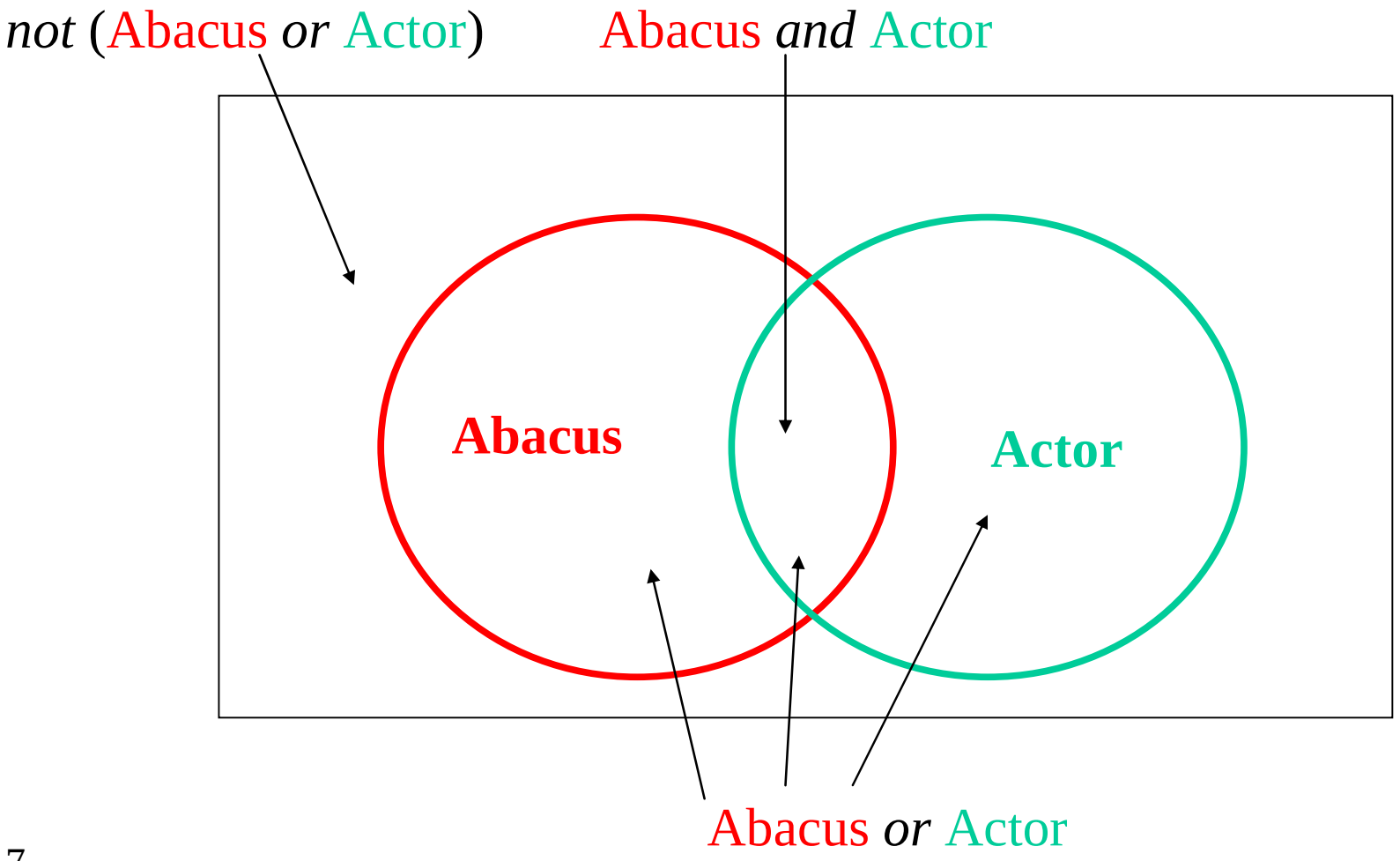
Inverted file (con file di posting semplice):



Lista invertita di abacus

Le parole sono i termini indice selezionati nella fase di indicizzazione

Valutazione di query
Diagramma Booleano



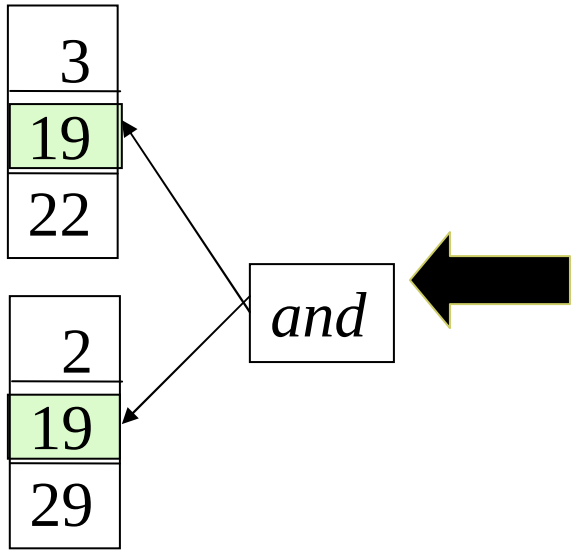
Valutazione di una query Booleana

Si accede al file dizionario (mediante ricerca binaria se indice lineare). Si recupera la lista di posting corrispondente al termine

Esempio: abacus and actor

Lista invertita di *abacus*

Lista invertita di *actor*



Il meccanismo di matching costruisce un albero binario di valutazione della query Booleana

Il documento 19 è l'unico che contiene entrambi i termini *abacus* e *actor*

Ordine di valutazione

L'ordine di valutazione di query Booleane è importante e deve essere specificato:
Es:

posting energia = d1, d3, d5, d7
posting nucleare = d2, d3, d4, d5, d6
posting solare = d4, d6, d8

Q= energia AND nucleare OR solare

Da sinistra
{d3, d5} $\cup$ {d4, d6, d8}= {d3, d5,d4, d6, d8}
Da destra:
{d2,d3,d4,d5,d6,d8} $\cap$ {d1, d3, d5, d7}={d3,d5}

Ordine di valutazione di operatori Booleani

Definizione delle regole di Precedenza:

adj, *near*
and, *not*
or

alta

bassa

Esempio

A *and* B *or* C *and* B

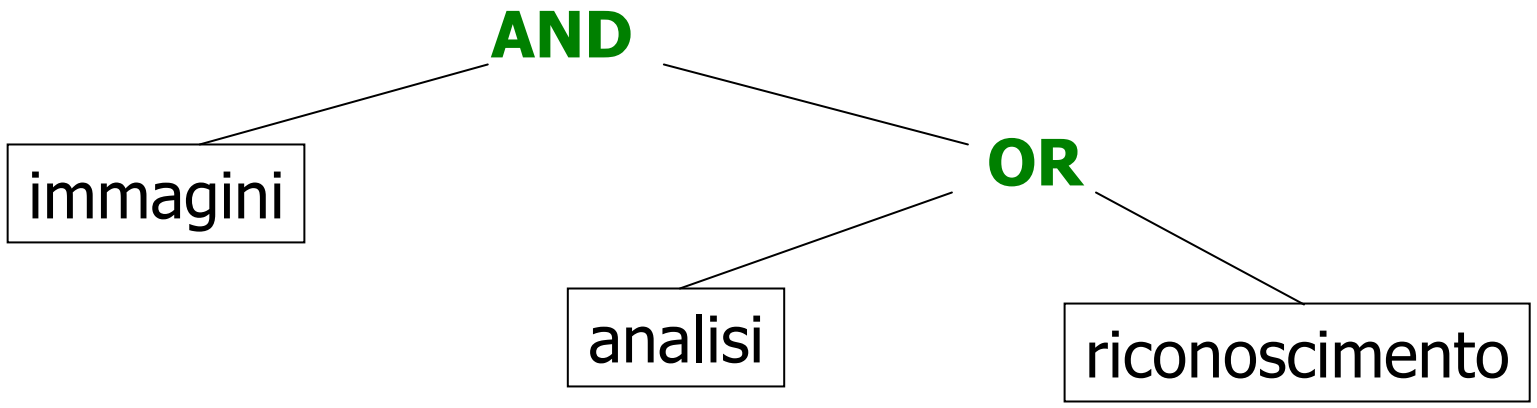
è valutata come

(A *and* B) *or* (C *and* B)

Valutazione di query Booleane

Costruzione dell'Albero binario di valutazione

Query:= immagini AND (analisi OR riconoscimento)



Algoritmo ricorsivo

Query Booleane

- Keywords combinate tramite operatori Booleani
  - OR: (e_1 OR e_2)
  - AND: (e_1 AND e_2)
  - NOT: (e_1 AND NOT e_2) puo' essere espresso come e_1 BUT e_2
- La Negazione viene generalmente implementata solo usando BUT per permettere un uso dell'indice inverted efficiente semplicemente filtrando un insieme reperito.
- Gli utenti inesperti hanno spesso problemi con la logica Booleana

Valutazione di query Booleane

- keyword: reperimento di tutti i documenti che contengono una keyword corrispondente a una foglia attraverso il file inverted e costruzione della lista di documenti corrispondente
- OR: costruzione di una lista associata al nodo che è l'unione delle liste dei sottoalberi destro e sinistro.
- AND: costruzione di una lista associata al nodo che è l'intersezione delle liste dei sottoalberi destro e sinistro.
- BUT = AND NOT: costruzione di una lista associata al nodo che è la differenza tra le liste dei sottoalberi destro e sinistro

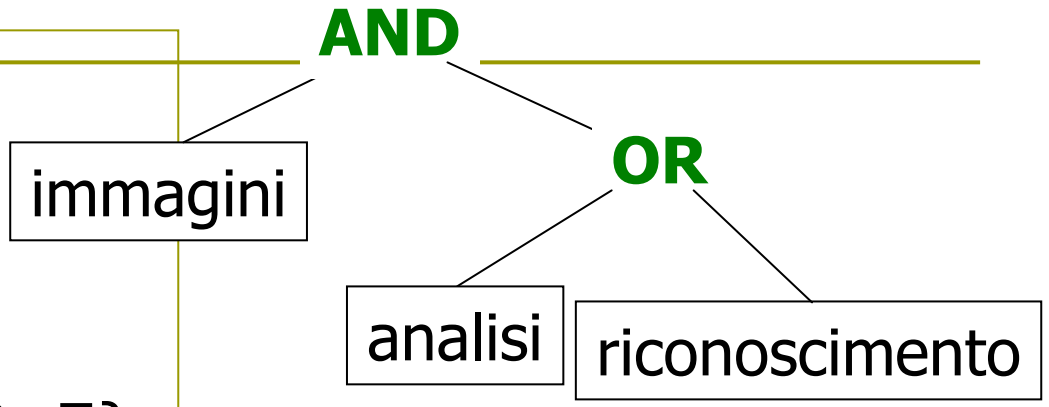
Valutazione di query Booleane

Inverted index

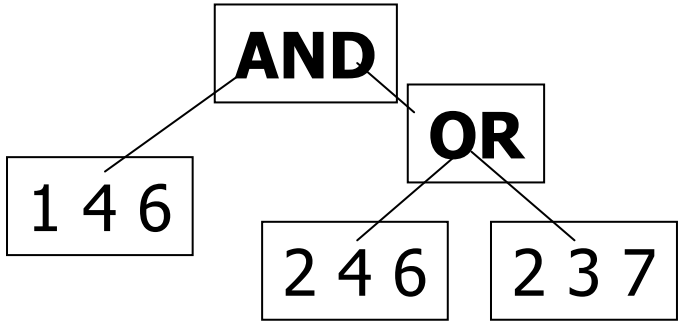
Immagini:={1, 4, 6}

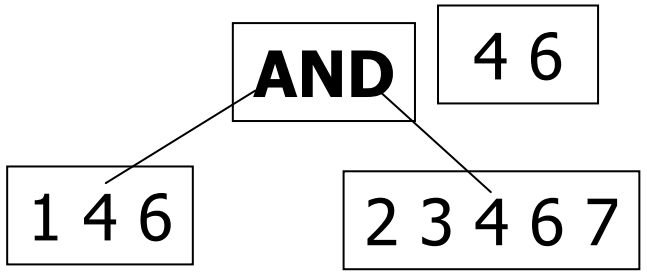
Analisi:={2, 4, 6}

Riconoscimento:={2, 3, 7}



Full evaluation mode





Visita ricorsiva in post-ordine:
Si **allocano in memoria liste intermedie** per i risultati dei nodi

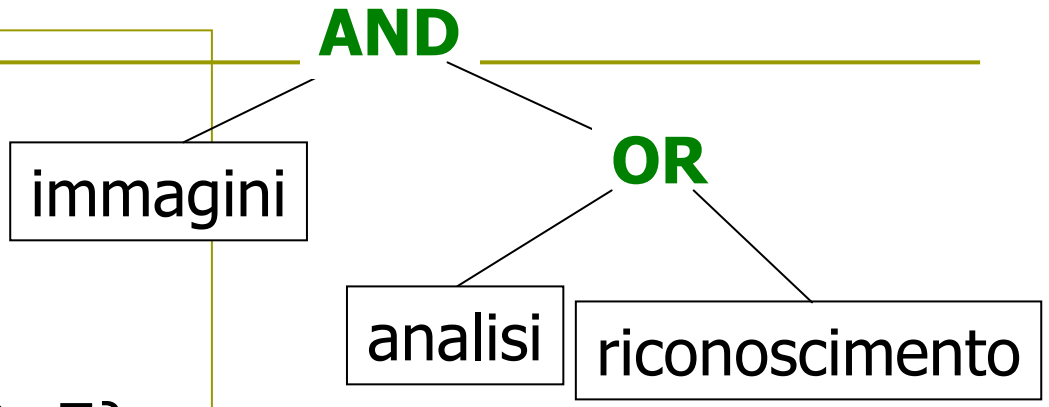
Valutazione di query Booleane

Inverted index

Immagini:={1, 4, 6}

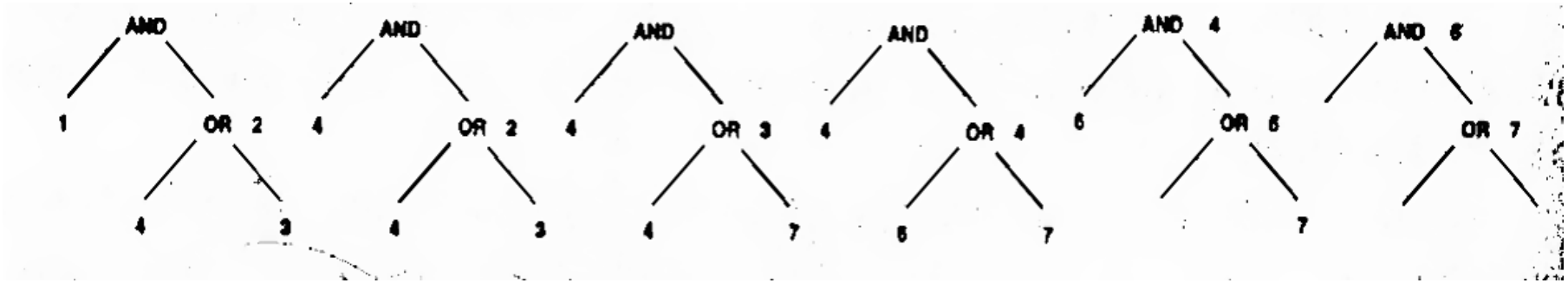
Analisi:={2, 4, 6}

Riconoscimento:={2, 3, 7}



Lazy evaluation mode

Non si allocano liste per i risultati parziali dei nodi



Ottimizzazione:valutazione di query Booleane

Q=pippo and pluto and paperino

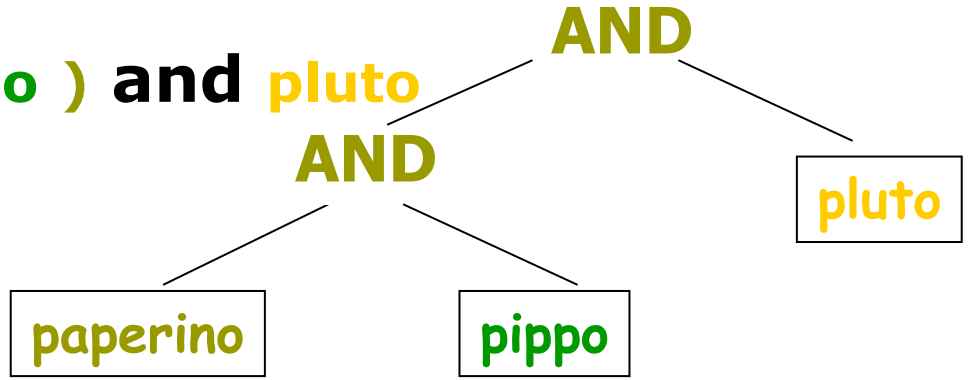
Dizionario	Posting list
pippo, 8	1, 2, 4, 11, 31, 45, 173, 174
pluto ,25	1, 2, 4, 5, 6, 16, 57, …,….,…,…
paperino ,4	2, 31, 54, 101

Come organizzare la valutazione per compiere meno lavoro?

→ Scelta dell'ordine di valutazione

Es: In ordine di frequenza crescente: utile tenere nel dizionario la frequenza totale dei termini

Q=(paperino and pippo) and pluto



Ottimizzazione:valutazione di query Booleane

Q=(pippo or pluto) and (paperino or topolino)

If #pippo + #pluto < #paperino + # topolino

→Q=(pippo or pluto) and (paperino or topolino)

Else →Q=(paperino or topolino) and (pippo or pluto)

Q=(paperino and not topolino)

If #topolino << #D → # not topolino >> # paperino

→ Q=(paperino) and not topolino

Inconvenienti del Modello Booleano di IR

- Il Retrieval è basato su un criterio decisionale binario (confronto esatto)
  - Non è in grado di produrre un ordinamento dei risultati:

    Q= a and b and c and d and e
  - Vengono esclusi sia d_1={a} sia d_2={a,b,c,d}

    Q= a or b or c or d or e
  - rank d_1={a} uguale al rank di d_2={a,b,c,d,e}

- Le query Booleane formulate dagli utenti sono spesso troppo semplicistiche e ambigue
  - (and linguistico → or logico) (difficile l'uso di parentesi)
- Non si può controllare il risultato: produce come risultato o troppi, o nessun documento

Il modello Vettoriale di IR di G. Salton

La rilevanza è modellata come *proprietà graduale* dei documenti → *ordinamento dei risultati* in funzione decrescente di rilevanza rispetto alla query.

E' basato *sull'algebra lineare* e rappresenta sia i documenti sia le query in uno spazio vettoriale n-dimensionale, ove n è il numero totale di termini indice

Un documento d_j è rappresentato mediante un vettore di pesi w_{kj} : $R(d_j) := \vec{d_j} = (w_{1j}, w_{2j} \ldots w_{Ni})$

i pesi possono essere sia valori in {0,1} sia valori numerici positivi.

Una query viene generalmente espressa come una lista di parole ed è rappresentata da un vettore di pesi che valgono 1 per le parole contenute e 0 per quelle non presenti:

$$R(q) := \vec{q} = (w_{1q}, w_{2q} \ldots w_{Nq})$$

Il modello Vettoriale di IR di G. Salton

Assunzioni

- La ***rilevanza è un concetto graduale*** ed è proporzionale alla similarità tra il vettore che identifica un documento e il vettore che identifica la query:

$$\text{Rilevanza}(d) \approx \text{sim}(\vec{d}, \vec{q})$$

- **Indipendenza reciproca dei termini:** La presenza contemporanea di coppie o di più termini nei documenti non è correlata in alcun modo

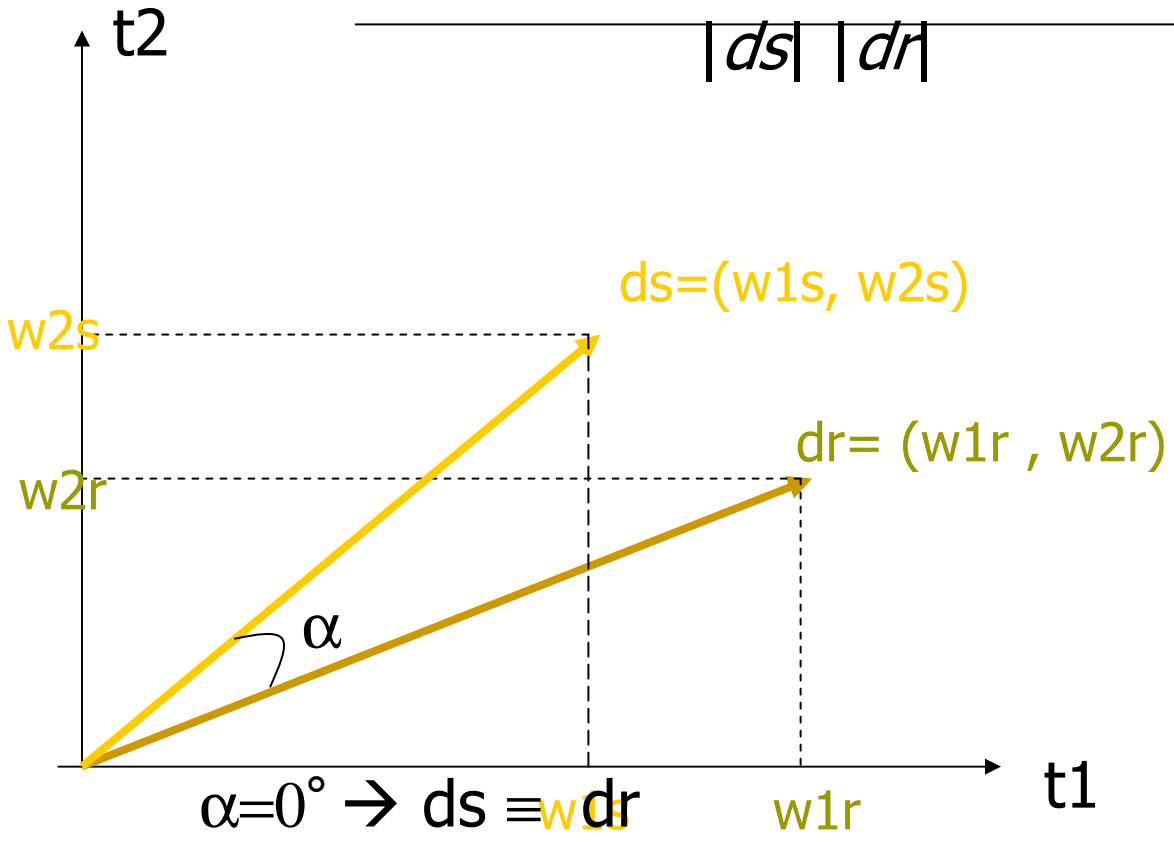
Il modello Vettoriale di IR

- *Rappresentazione basata su termini indice di*
  documenti $d_i=(w_{1i}, w_{2i} \ldots, w_{ni})$ → Vettore documento
  query $q_q=(q_{1q}\ q_{2q}, \ldots, q_{Lq})$ → Vettore query
- *T* termini distinti sono usati per caratterizzare il contenuto.
- Ogni termine i è identificato da un **vettore base** dello spazio $\vec{t_i}$ **= [0, 0, 0,..., 1, ..., 0]**
- *I* vettori t sono linearmente indipendenti e formano una **base ortonormale** per lo spazio *N* dimensionale
- Ogni vettore nello spazio (sia vettore documento sia vettore query) è una combinazione lineare di *N* vettori termini.
- L' *resimo* documento d_r può essere rappresentato come un vettore documento :

$$\vec{d}_r = \sum_{i=1}^{N} w_{ir}\,\vec{t}_i$$

Rappresentazione di documenti nello spazio bidimensionale definito da due termini

- Prodotto tra due vettori $x \bullet y = |x|\,|y| \cos\alpha$
- similarità tra vettori $\cos\alpha = (x \bullet y)/|x|\,|y|$
- $\cos\alpha = \dfrac{(w1s\ w1r\ t1 \bullet t1 + w1s\ w2r\ t1 \bullet t2 + w2s\ w1r\ t2 \bullet t1 + w2s\ w2r\ t2 \bullet t2)}{|ds|\,|dr|}$



Misura di similarità

Vettore documento:

$$\vec{d}_r = \sum_{i=1}^{N} w_{ir} \vec{t}_i$$

Vettore query:

$$\vec{q} = \sum_{j=1}^{L} q_j \vec{t}_j$$

$$\vec{d}_r \bullet \vec{q}_s = \sum_{i,j=1}^{N} w_{ir} q_j \vec{t}_i \bullet \vec{t}_j$$

Misura di similarità

- Il fattore di correlazione tra termini non è noto
  $t_i \bullet t_j$
- assunzione: i vettori dei termini sono una base ortonormale:

$$\vec{t}_i \bullet \vec{t}_j = 0 \ (i \neq j)$$
$$\vec{t}_i \bullet \vec{t}_j = 1 \ (i = j)$$

- Si assume che i termini non siano correlati
- quindi:

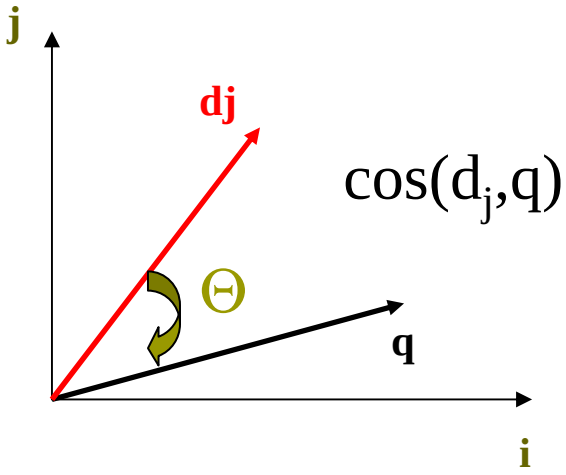
$$\vec{d}_r \bullet \vec{q} = \sum_{i=1}^{N} w_{ir} \times q_i$$

- Analogamente

$$\vec{d}_r \bullet \vec{d}_s = \sum_{i=1}^{N} w_{ir} \times w_i$$

Il Modello Vettoriale di IR

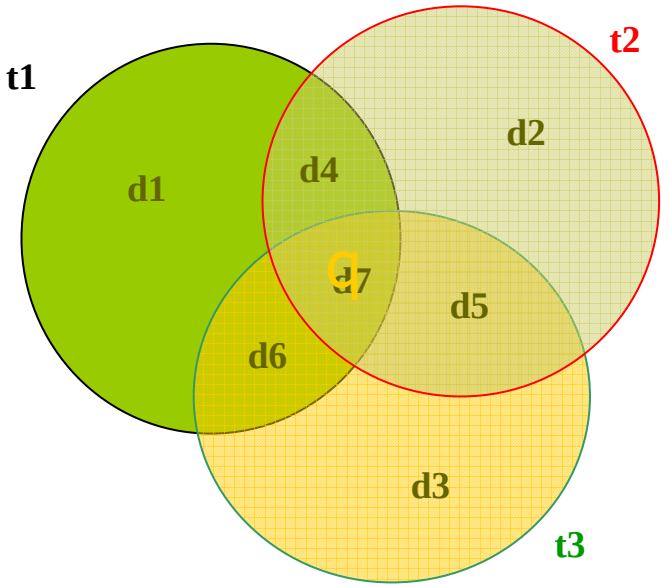
$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^{N} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{L} w_{i,q}^2}}$$



se $w_{ij} > 0$ and $w_{iq} > 0$, $0 <= sim(q,d_j) <= 1$

- Un documento viene reperito anche se non soddisfa completamente la query

Il Modello Vettoriale
Esempio I:
pesi binari nei documenti e nelle query



q= t1 t2 t3 → (1 1 1)

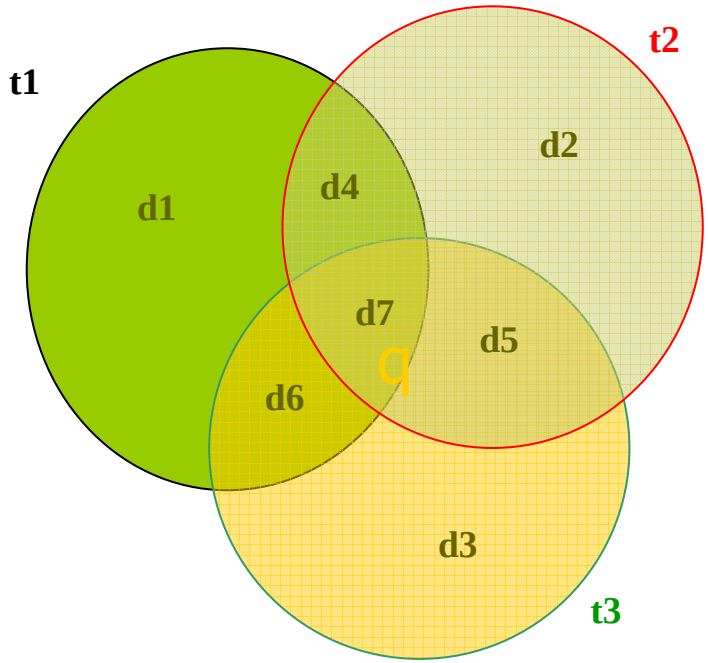
	t1	t2	t3		d*q	cos(d,q)
d1	1	0	0		1	0,57735
d2	0	1	0		1	0,57735
d3	0	0	1		1	0,57735
d4	1	1	0		2	0,816497
d5	0	1	1		2	0,816497
d6	1	0	1		2	0,816497
d7	1	1	1		3	1
q	1	1	1		3	1

Il Modello Vettoriale di IR

I **pesi binari sono troppo limitanti**

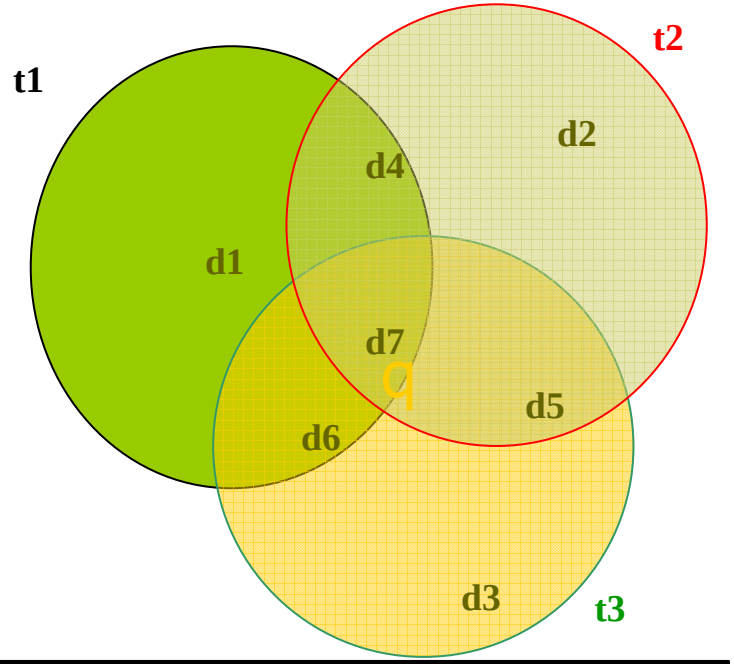
- $w_{ij} > 0$ quando t_i *appartiene a* d_j
- $w_{iq} >= 0$ *è associato alla coppia* (t_i, q)
- $\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{tj})$ $\vec{q} = (w_{1q}, w_{2q}, \ldots, w_{tq})$
- Ad ogni termine t_i è associato un vettore unitario $\vec{t_i}$
- I vettori unitari $\vec{t_i}$ e $\vec{t_j}$ sono considerati ortonormali (i.e., si assume che i termini indice appaiano indipendentemente gli uni dagli altri nei documenti)
- I t vettori unitari $\vec{t_i}$ formano una base ortonormale per uno spazio t-dimensionale dove query e documenti sono rappresentati come vettori pesati

Il Modello Vettoriale
Esempio II:
pesi binari nei documenti



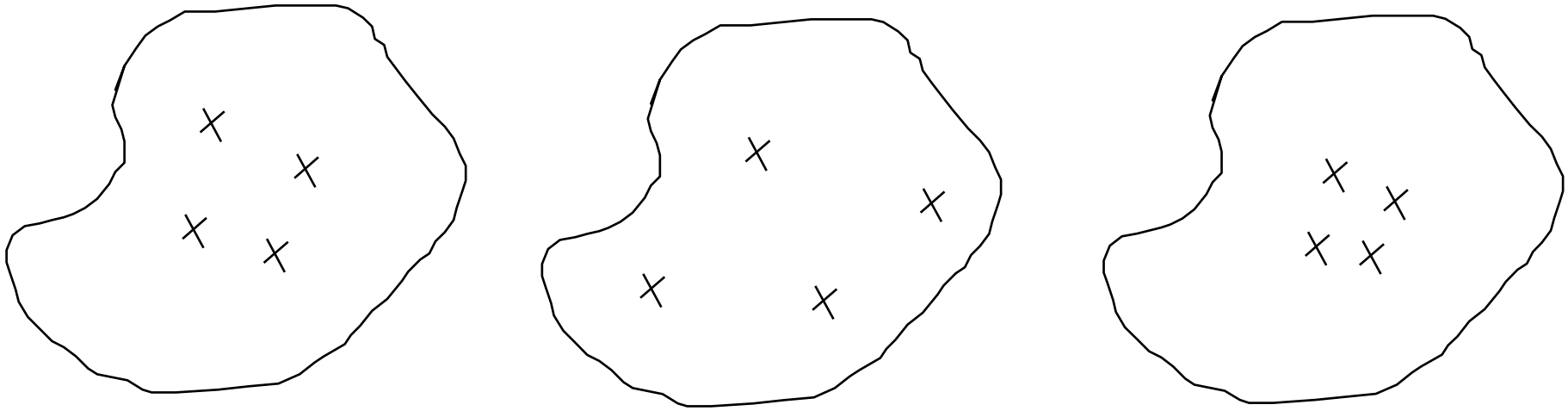
	t1	t2	t3	|d| |q|	d*q	cos(d,q)
d1	1	0	0	1*3.464	1	0,267261
d2	0	1	0	3.464	2	0,534522
d3	0	0	1	3.464	3	0,801784
d4	1	1	0	4,898	3	0,566947
d5	0	1	1	….	5	0,944911
d6	1	0	1		4	0,755929
d7	1	1	1		6	0,92582
q	1	2	3		14	1

Il Modello Vettoriale di IR: Esempio III termini e query pesate



	t1	t2	t3		d*q	cos(d,q)
d1	1	0	0		1	0,267261
d2	0	2	0		4	0,534522
d3	0	0	3		9	0,801784
d4	1	2	0		5	0,597614
d5	0	1	2		8	0,956183
d6	1	0	2		7	0,83666
d7	1	1	2		9	0,981981
q	1	2	3		14	1

Spazio virtuale dei documenti: influenza del grado di significatività



Stato originale

Dopo l’assegnamento di termini con buon potere discriminante

Dopo l’assegnamento di termini con poco potere discriminante

Il Modello di IR Vettoriale: definizioni alternative di similarità

$$\mathbf{sim(d_i, q)} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{w_{ki} w_{kq}}}{\sqrt{\sum_{\mathbf{k=1}}^{\mathbf{N}} \mathbf{w_{ki}^2} \sum_{\mathbf{k=1}}^{\mathbf{L}} \mathbf{w_{kq}^2}}}$$

Coseno

$$sim(d_i, q) = \frac{2 \sum_{i=1}^{N} w_{ki} w_{kq}}{\sum_{k=1}^{N} w_{ki}^2 \sum_{k=1}^{L} w_{kq}^2}$$

Coefficiente Dice

$$sim(d_i, q) = \frac{\sum_{i=1}^{N} w_{ki} w_{kq}}{\sum_{k=1}^{N} w_{ki}^2 + \sum_{k=1}^{L} w_{kq}^2 - \sum_{i=1}^{N} w_{ki} w_{kq}}$$

Coefficiente Jaccard

normalizzazione basata sulla lunghezza del documento

Il Modello di IR Vettoriale

Vantaggi

- La pesatura dei termini migliora la qualità del risultato
- Il confronto parziale permette di reperire documenti che approssimano le condizioni espresse nella query
- la formula del coseno permette di ordinare i documenti in funzione del grado di similarità alla query

Svantaggi

- L'assunzione dell'indipendenza tra i termini non ha fondamento;
- I termini che non sono presenti nelle query non dovrebbero avere influenza sul retrieval
- Il liguaggio di query non è abbastanza espressivo

Esempio: Implementazione del modello Vettoriale di IR

doc	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
D_1	0	0	.477	0	.477	.176	0	0	0	.176	0
D_2	0	.176	0	.477	0	0	0	0	.954	0	.176
D_3	0	.176	0	0	0	.176	0	0	0	.176	.176
Q	0	0	0	0	0	.176	0	0	.477	0	.176

Calcolo del ranking con il prodotto scalare

Sim(Q, D1) = (0)(0) + (0)(0) + (0)(0.477) + (0)(0) + (0)(0.477)+ (0.176)(0.176) + (0)(0) + (0)(0)=0.031

Sim(Q, D2)=0,486 Sim(Q,D3)=0,062

ranking D2, D3, D1

SIRE : Un IRS Booleano con Ranking vettoriale

- **Query: Booleane**

Identifica l'insieme di documenti che soddisfano la query Booleana

A AND B OR C

- Ordina tali documenti adottando un criterio di ranking basato sul modello vettoriale
  - (prodotto interno)

A B C

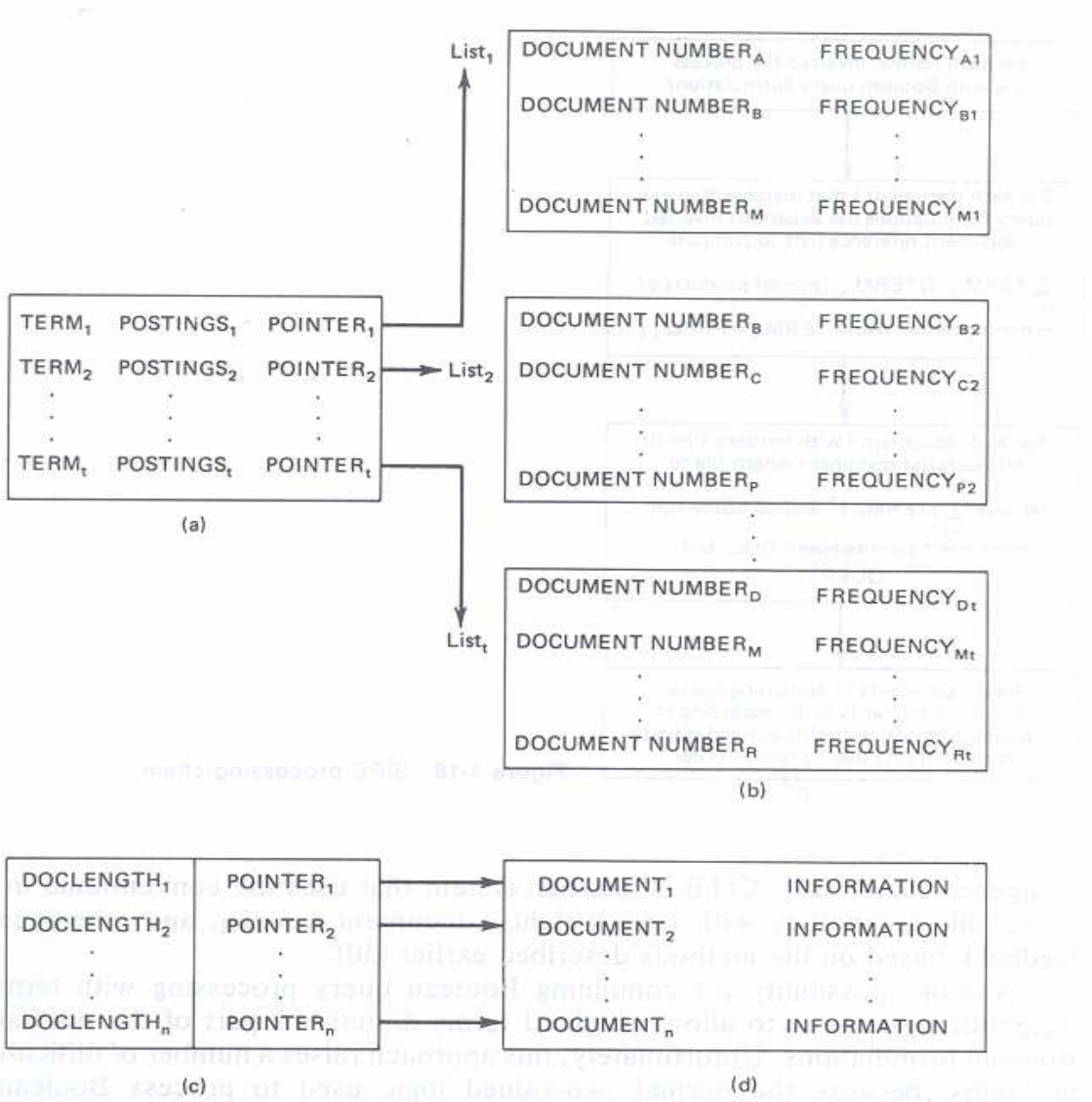


Figure 4-15 SIRE file organization. (a) Directory to inverted lists. (b) Inverted document reference lists. (c) Document length list. (d) Main document file.

Breve confronto dei modelli classici

- Il Modello Booleano di IR non fornisce un ordinamento dei risultati
- Salton and Buckley hanno valutato il modello vettoriale di IR e concluso che con collezioni generiche è migliore del modello probabilistico di IR
- Questa stima è a tutt'oggi condivisa

Linguaggi di Query per Sistemi di IR basati sui modelli di base

- TIPOLOGIE
  - Linguaggi per utenti
    - query Booleane
    - Query sulla Struttura dei documenti
    - query contestuali (Frasi & Prossimità)
    - Query vettoriali
    - query Booleane pesate

Frasi

- Reperimento dei documenti contenenti una frase (definita come lista ordinata di parole contigue)
  - "information theory"
- E' possibile che nel testo del documento siano state rimosse le stopwords e sia stato effettuato lo stemming.
  - "buy camera" è soddisfatta da:
    "buy a camera"
    "buying the cameras"
    etc.

Valutazione di Frasi

- E’ necessario che il file inverted memorizzi le posizioni di ogni keyword del documento.
- Si reperiscono i documenti e le posizioni delle singole parole nella frase, e quindi si controlla la contiguità delle posizioni.
- **E’ meglio iniziare il controllo dalle parole meno frequenti nella frase.**

Operatori di adiacenza e di vicinanza

abacus *adj* actor

Si richede che abacus e actor siano adiacenti come nella stringa

"abacus actor".

abacus *near 4* actor

Si richede che abacus e actor siano vicini come nella stringa "abacus utilized by actor".

Alcuni sistemi supportano query del tipo

with (due termini nella stessa frase) o

same (due termini nello stesso paragrafo).

Valutazione di operatori di Adiacenza

Esempio: abacus *adj* actor

Postings per abacus

3	94
19	7
19	212
22	56

Postings per actor

3	66
19	213
29	45

Gli operatori con contesto si valutano in modo efficiente effettuando prima una query con *l'and*

E solo per i documenti che la soddisfano si valuta l'operatore *adj* o *near*

Documento 19, posizioni 212 e 213, sono le sole occorrenze di "abacus" e "actor" adiacenti

Algoritmo per la Valutazione di Frasi

- Valuta la query mettendo in AND i termini della frase (t_t...AND...t_m) e identifica con A l'insieme dei documenti che la soddisfa
- Inizializza a vuoto l'insieme *R* dei documenti reperiti
- Per ogni documento *d* in *A*
- individua il vettore P_i di posizioni delle occorrenze di ogni termine in *d*
- Trova il vettore più corto P_s dei P_i i=1,…,m
- Per ogni posizione (occorrenza) *p* del termine t_s in P_s
  Per ogni termine t_i diverso da t_s
- effettua la ricerca binaria per trovare una posizione ($p - s + i$) nel vettore P_i
- Se la posizione è trovata per ogni termine, aggiungi *d* a *R*
- Return *R*

Valutazione di query con operatori di prossimità

- Identifica i documenti che contengono tutte le parole.
- Usa un approccio simile a quello per la ricerca di frasi.
  - Durante la ricerca binaria ricerca le posizioni delle parole rimanenti e trova la posizione più vicina di t_i a *p* e controlla che la vicinanza sia inferiore al massimo valore specificato dall'operatore.

Query sulla struttura dei documenti

- Si assume che i documenti abbiano una struttura che possa essere sfruttata nella ricerca.
- La struttura potrebbe essere:
  - Un insieme fisso di campi, es. *titolo, autori, abstract*, ecc..
  - Gerarchica di campi (struttura ad albero):

  - Ipertesto.

Query sulla struttura di documenti

- Le query permettono di esprimere:
  - **condizioni sul *contenuto*:** tramite keywords o espressioni regolari e combinazioni con op. Booleani
  - **condizioni sulla struttura**: esistenza di specifici campi, contenimento delle condizioni sul contenuto in specifici campi, contenimento e prossimità tra campi
    - es: doc. contenenti il campo "**abstract**"
    - doc. contenenti "*nuclear fusion*" in un **titolo** di **capitolo**
    - es: Es: doc. Contenenti "*nuclear fusion*" nella stessa sezione
- Le condizioni sulla struttura sono associate alle componenti atomiche della query Booleana, cioè i termini di ricerca, e vengono quindi valutate nelle foglie dell'albero binario della query.

Query sulla struttura e contenuto dei documenti

Estensione del file inverted con i campi

- inclusione dell'ID delle sezioni o campi

  information: doc1.{sez1, sez2, sez3, …}
  retrieval: doc1.{sez1, sez3, …}

  Permette di valutare

- **Vincoli di inclusione**
  - A in sezione S
  - A nella stessa sezione di B

    I termini A e B devono co-occorrere in una sezione comune

Estensione del file inverted per valutare query su struttura

- Vincoli di prossimità
  - esempio di query
    (information adiacente retrieval)
    (information within 5 words retrieval)
- Inclusione del
  - numero di paragrafo
  - numero di frase nel paragrafo
  - numero di parola nella frase
  - Si identificano i paragrafi e le frasi individuando nel testo dei caratteri specifici

Estensione del file inverted

- **inclusione della posizione nei paragrafi delle sezioni**
- **information:** **doc1.{sez1,1, sez2,1, sez3,3, …}**
- **retrieval:** **doc1.{sez1,3, sez2,2, sez3,3, …}**

- **inclusione della posizione nelle frasi dei paragrafi**
- **information:** **{sez1,1,30 sez2,1,{14,20} sez3,3,{14,24} …}**
  retrieval: **{sez1,3,1 sez2,2,23 sez3,3,24 …}**

- costo: dimensione degli indici
- Limitazione: si reperisce l'intero documento