Sistemi per il recupero delle informazioni

Gabriele Pozzani

A.A. 2012/2013

Corso di Laurea Magistrale in Editoria e Giornalismo

SRI: tipi di interrogazioni

Le interrogazioni nei SRI (1)

- I diversi SRI possono permettere la formulazione di diversi tipi di query:
  - Booleane
    - Prime
    - Più diffuse
  - Vettoriali
  - Booleane estese
  - Fuzzy
  - Probabilistiche
  - Linguaggio naturale

Le interrogazioni nei SRI (2)

- Ogni tipo di query è caratterizzato da:
  - Un diverso modo (modello) per rappresentare i documenti e le query
  - Un diverso metodo di ordinare (ranking) i documenti recuperati per presentarli all'utente

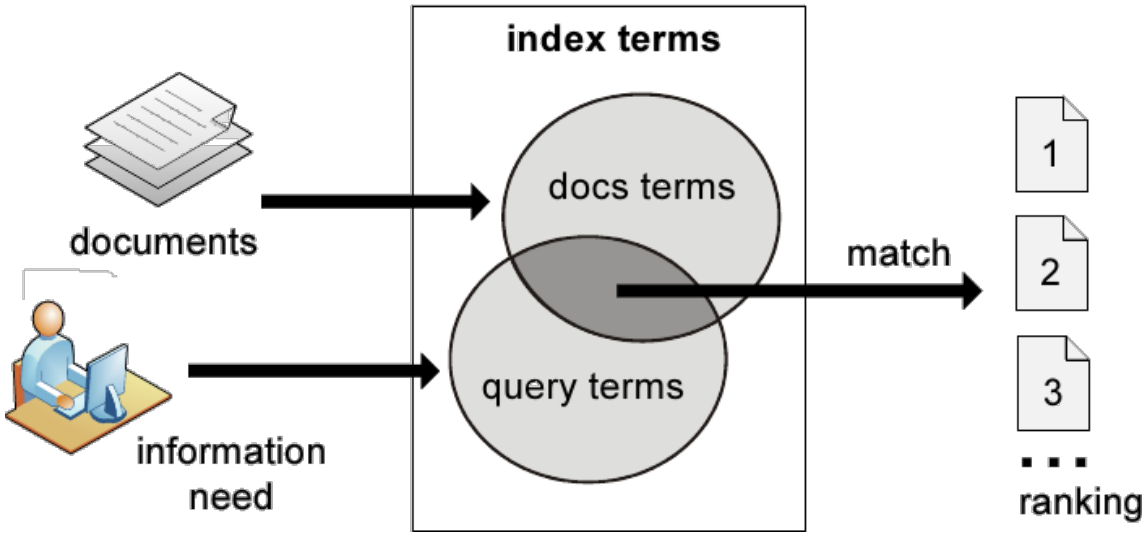
Le interrogazioni nei SRI (3)

- Solitamente i SRI sono basati sull'uso di “termini di indicizzazione”
  - Sono parole chiave
    - In generale sono le parole che compaiono nei documenti “interessanti”
  - Le interrogazioni sono “composte” di termini d'indicizzazione
    - Semplifica la scrittura di una query

Le interrogazioni nei SRI (4)

- Il ranking è l'ordinamento dei documenti recuperati e ritornati all'utente
  - Riflette la rilevanza rispetto all'interrogazione dell'utente
  - “predire” la rilevanza dei documenti

Processo di recupero delle informazioni



Termini d'indicizzazione

- Ogni documento è rappresentato da un insieme di termini d'indicizzazione o parole chiave
- Un termine d'indicizzazione può essere
  - Una parola
  - Una sequenza di parole consecutive in un documento
  - Quando tutte le parole dei documenti sono usate come termini d'indicizzazione si parla di ricerca full-text

Vocabolario

- L'insieme di tutti i termini d'indicizzazione costituisce il vocabolario
  - $t_1, t_2, t_3, \ldots, t_k$

Rappresentazione matriciale: termine-documento

- L'occorrenza di un termine t_i in un documento d_i mette in relazione t_i con d_i
  - matrice “termine-documento”

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_M \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & \ldots & d_N \end{array} \\ \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,N} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M,1} & f_{M,2} & \cdots & f_{M,N} \end{bmatrix} \end{array}$$

  - $f_{k,h}$ rappresenta la frequenza del termine t_k nel documento d_h

Interrogazioni Booleane

Le interrogazioni Booleane

- Le interrogazioni booleane sono le prime e più diffuse
  - Usate nei motori di ricerca
- Formula booleana: lista di termini booleani (sempre o Veri o Falsi) connessi da operatori booleani
  - and (∧)
  - or (∨)
  - not (¬)
- In una query, un termine rappresenta la presenza o meno di quel termine nei documenti cui l'utente è interessato
  - Esempio: *ristorante* ∧ *(cinese* ∨ *thailandese)* ∧ ¬*pizzeria*

Operatori booleani

- and (∧)
  - A *and* B: il risultato è vero se e solo se sono veri sia A che B
- or (∨)
  - A *or* B: il risultato è vero se e solo se almeno uno tra A e B è vero
- not (¬)
  - *not* A: il risultato è vero se A è falso e viceversa.
- Xor (⊕)
  - A xor B: il risultato è vero se e solo se solo uno tra A e B è vero

Tabelle di verità

- Le tabelle di verità elencano tutte le possibili combinazioni di valori degli operandi e il corrispondente valore del risultato
  - Il Vero si indica anche con l'1
  - Il Falso si indica anche con lo 0

A	B	A *and* B	A *or* B	*not* A	A *xor* B
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

Recupero dei documenti

- Un documento è recuperato se soddisfa l'interrogazione, viene scartato altrimenti
  - Un documento o è rilevante o è non rilevante
    - Nessuna via di mezzo

Svantaggi (1)

- Non è possibile pesare i termini per dare loro importanze diverse
  - Un termine è presente o assente
  - Ad esempio non è possibile rispondere alla query “musica di Beethoven, preferibilmente una sonata”
- Altri tipi di interrogazioni cercano di risolvere questo problema

Svantaggi (2)

- Non essendo i termini pesati, nemmeno i documenti possono essere valutati più o meno rilevanti
  - Non è possibile alcun tipo di ranking dei documenti recuperati

Svantaggi (3)

- Le richieste vanno trasformate in un'interrogazione booleana, ma l'utente non ha sempre ben chiaro il significato degli operatori booleani
  - Errori nella formulazione delle query
- Spesso l'utente pone l'interrogazione interpretando gli operatori logici in funzione del contesto e dei termini
  - Caffè and Brioche or Muffin
  - Impermeabile and Ombrello or Occhiali da sole
- L'uso delle parentesi riduce il problema
  - Caffè and (Brioche or Muffin)
  - (Impermeabile and Ombrello) or Occhiali da sole

Svantaggi (4)

- Sistemi di IR diversi possono usare ordini di precedenza degli operatori booleani diversi
  - Gli operatori vengono valutati nell'ordine not, and, or con precedenza da sinistra a destra per quelli dello stesso tipo
  - Gli operatori vengono valutati seguendo strettamente l'ordine da sinistra a destra senza considerare il tipo di operatore
- Le parentesi vengono valutate come un'unità prima di essere combinati con le parti fuori dalle parentesi
- L'uso delle parentesi risolve il problema

Google usa il modello booleano? (I)

- In Google l'interpretazione di una query [t_1 t_2 t_3…t_k] è [t_1 AND t_2 AND t_3 AND … AND t_k]
- In alcuni casi però alcuni risultati non contengono un termine t_i
  - Una pagina contiene una variante (morfologica, sinonimica, correzione ortografica) di t_i
  - Query lunghe (k molto grande)
  - Vi sono pochissimi risultati
- Google permette anche gli altri operatori Booleani
  - OR *[ristorante cinese OR thai]*
  - NOT *[ristorante -cinese]*

Google usa il modello booleano? (II)

- Google fornisce anche altri operatori non Booleani
  - Inclusione di parole simili *[~ristorante]*
  - Ricerca all'interno di un sito *[Olimpiadi site:.gov]*
  - Riempi lo spazio vuoto
    *["un * risparmiato è un * guadagnato"]*
- Il modello booleano non ordina in modo particolare i risultati
  - Google invece ordina i risultati (in base ad una qualche misura) dal migliore al peggiore
    - Come fa? Va oltre il semplice modello booleano...e vi andremo anche noi ;)

Interrogazioni Vettoriali

Le interrogazioni vettoriali

- Superano alcuni limiti delle interrogazioni booleane
- Assegnano dei valori/pesi non binari ai termini nelle query e nei documenti
  - I pesi sono usati per calcolare la similarità tra la query e i documenti
  - Il ranking si ottiene ordinando i documenti in ordine decrescente di grado di similarità

Esempio di query vettoriale

$$q = (ristorante_2, cinese_{0.8}, thailandese_{0.8}, pizzeria_{0.2})$$

Documenti e interrogazioni come vettori

- Query e documenti sono rappresentati da vettori
  - $d_j = (w_{1j}, w_{2j}, \ldots, w_{Mj})$
  - $q = (w_{1q}, w_{2q}, \ldots, w_{Mq})$
- w_{kh} rappresenta il peso del termine t_k nel documento d_h (o nella query q)
- I pesi dei termini nei documenti possono essere calcolati con varie formule, la più utilizzata nei sistemi odierni è TF–IDF:
  - TF (Term-Frequency)
  - IDF (Inverse Document Frequency)

TF-IDF

$$w_{kh} = tf_{kh} * idf_k = \frac{\#\text{ occorrenze } t_k \text{ in } d_h}{\#\text{ docs in cui } t_k \text{ occorre}}$$

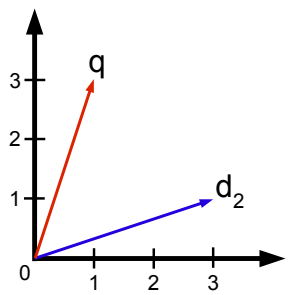
- tf_{kh} = numero di occorrenze del termine t_k nel documento d_h
  - più un termine compare in un documento, più è importante per descrivere quel documento
- idf_k = inverso della frequenza del termine t_k nella collezione di documenti = $\log(N/n_k)$ dove N è il numero di documenti della collezione e n_k è il numero di documenti che contengono il termine t_k
  - più sono i documenti in cui appare un termine, meno quel termine è importante (per discriminare i documenti)

Documenti come vettori

- Ad ogni documento viene associato il vettore contenente i pesi dei termini in esso
  - Supponiamo che il vocabolario contenga solo i termini "ristorante" e "pizzeria"
  - Per ogni termine si calcola il suo peso in ogni documento tramite TF-IDF
  - Supponiamo che il peso di "ristorante" e "pizzeria" in d_1 sia rispettivamente 100 e 300
    Supponiamo che il peso di "ristorante" e "pizzeria" in d_2 sia rispettivamente 3 e 1
  - $d_1 = (100, 300)$
    $d_2 = (3, 1)$

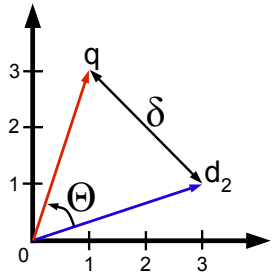
Similarità tra documenti e query (1)

- La similarità tra i documenti e una query corrisponde alla similarità tra i vettori che li rappresentano
  - Ogni vettore può essere rappresentato in uno spazio (sul piano cartesiano nel caso bidimensionale)



Similarità tra documenti e query (2)

- La similarità tra i vettori può essere calcolata in diversi modi
  - Distanza euclidea δ: un doc è tanto più simile alla query quanto contiene gli stessi termini con gli stessi pesi
    - 0 → massima similarità
  - Misura del coseno Θ: un doc è tanto più simile alla query quanto contiene gli stessi termini in proporzioni simili
    - 1 → massima similarità
    - 0 → minima similarità



Similarità tra documenti e query (3)

- Esempio

  Q = (1, 3)

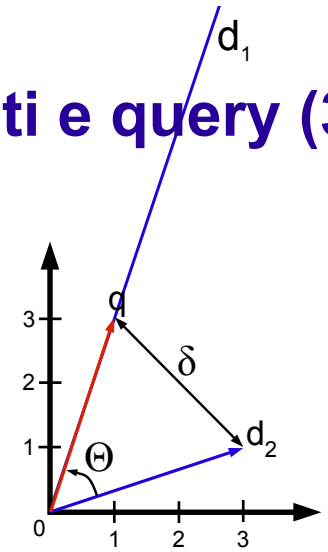
  D_1 = (100, 300)

  D_2 = (3, 1)

  - $\delta(Q,D_1) = 313{,}07$

    $\delta(Q,D_2) = 2{,}83$ → D_2 è più simile a Q con la distanza euclidea
  - $\Theta(Q,D_1) = 1$ → D_1 è più simile a Q con la misura del coseno

    $\Theta(Q,D_2) = 0{,}6$



Similarità tra documenti e query (4)

- L'esempio visto era con soli due termini per permetterne la facile rappresentazione su un piano
- Quando si hanno *n* termini si hanno vettori con *n* valori che rappresentano punti in uno spazio a *n* dimensioni
- La distanza euclidea e la misura del coseno possono essere definite su spazi n-dimensionali

Recupero dei documenti

- Viene calcolata la similarità di ogni documento con la query
- I documenti sono ritornati in ordine decrescente di similarità
  - Dal più simile al meno simile

Interrogazioni Booleane estese

Query booleane estese

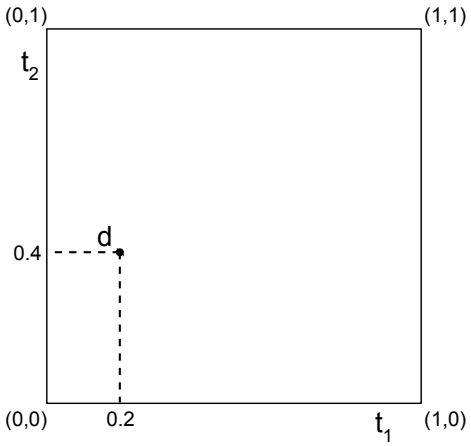
- Nelle query booleane non è possibile ordinare i documenti recuperati in base ad un ranking
  - Tutti i documenti sono ugualmente importanti per il sistema
- Estendiamo le query booleane combinando alcune caratteristiche delle query vettoriali
  - Pesatura dei termini
  - Matching parziale

Idea (1)

- I termini di ogni documento sono pesati
  - Usando ad esempio TF-IDF
- Una query booleana estesa è una normale query booleana
  - $q_{and} = t_1 \wedge t_2$
  - $q_{or} = t_1 \vee t_2$
- I documenti sono recuperati e ritornati in base al peso in essi dei termini nella query
- Ma come si tiene conto degli operatori booleani?

Idea (2)

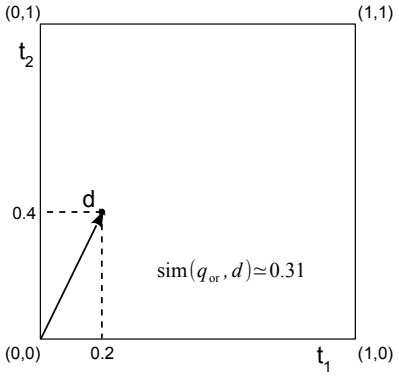
- Poniamo di avere solo due termini t_1 e t_2
  - allora come già visto possiamo rappresentare i documenti su un piano cartesiano
  - Esempio, in d
    - t_1 pesa 0.2
    - t_2 pesa 0.4



Idea (3): or

- Consideriamo la query $q_{or} = t_1 \vee t_2$
  - In un documento d più i due termini sono importanti più d soddisfa la query
  - Prendiamo il punto (0,0) del piano cartesiano come riferimento
    - La distanza del documento d dal punto (0,0) rappresenta la similarità di d con la query q_{or}

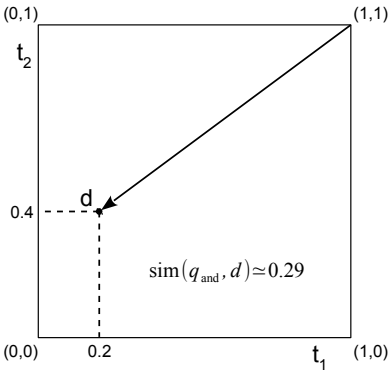
$$\mathrm{sim}(q_{\mathrm{or}}, d) = \sqrt{\frac{t_1^2 + t_2^2}{2}}$$



Idea (4): and

- Consideriamo la query $q_{and} = t_1 \wedge t_2$
  - La query è soddisfatta al massimo da un documento d quando in d entrambi i termini hanno peso 1
  - Prendiamo il punto (1,1) del piano cartesiano come riferimento
    - La distanza del documento d dal punto (1,1) rappresenta la similarità di d con la query q_{and}

$$\mathrm{sim}(q_{\mathrm{and}}, d) = 1 - \sqrt{\frac{(1-t_1)^2 + (1-t_2)^2}{2}}$$



Recupero dei documenti

- Viene calcolata la similarità di ogni documento con la query
- I documenti sono ritornati in ordine decrescente di similarità
  - Dal più simile al meno simile

Interrogazioni in linguaggio naturale

Query in linguaggio naturale

- Frontiera dell'information retrieval
- L'utente può porre un'interrogazione come una domanda nella propria lingua
  - q = “che ore sono?”
- Il sistema cerca di comprendere il significato della domanda e di rispondere
- Questo tipo di interrogazione è in generale
  - Imprecisa
  - Inaccurata

  come lo è una lingua
- Esempio “funzionante” più famoso: WolframAlpha
  - Esempio anche di sistema di “question answering”