Sistemi di Elaborazione dell'informazione II

Corso di Laurea Specialistica in Ingegneria Telematica
II anno – 4 CFU
Università Kore – Enna – A.A. 2009-2010

Alessandro Longheu
http://www.diit.unict.it/users/alongheu
alessandro.longheu@diit.unict.it

Dati Non Strutturati: Information Retrieval

Information Retrieval (IR): definizione

- **L'Information Retrieval** (IR) si occupa della rappresentazione, memorizzazione e organizzazione **dell'informazione non strutturata** (testuale), al fine di rendere agevole all'utente il soddisfacimento dei propri bisogni informativi.
- Data una collezione di documenti e un bisogno informativo dell'utente, lo scopo di un sistema di IR è di trovare informazioni che potrebbero essere utili, o rilevanti, per l'utente.
- Un esempio di "**bisogno informativo**": *trova tutti i documenti che contengono informazioni sui libri che sono stati scritti durante il 1800, o raccontano una storia d'amore e contengono riferimenti a città italiane.*
- Rispetto alla teoria classica delle basi di dati, l'enfasi non è quindi sulla ricerca di dati ma sulla ricerca di informazioni.
- Le tecniche di Information Retrieval, nate per dati testuali sono state in seguito estese ed applicate anche a dati multimediali (immagini, audio, video) e semi-strutturati.

Information Retrieval (IR): definizione

- Il settore dell'Information Retrieval è stato studiato fin dagli anni `70. Negli anni `90, l'esplosione del Web ha moltiplicato l'interesse per IR.
- Il Web infatti non è altro che un'enorme collezione di documenti, sui quali gli utenti vogliono fare ricerche.
- Il problema è che non è semplice caratterizzare esattamente i bisogni informativi dell'utente. Ciò deriva **dall'ambiguità e complessità** dei linguaggi naturali.

Information Retrieval vs Data Retrieval

- Un sistema di Data Retrieval (ad esempio un DBMS) gestisce dati che hanno una struttura ben definita.
- Un sistema di Information Retrieval gestisce testi scritti in linguaggio naturale, spesso non ben strutturati e semanticamente ambigui.
- Di conseguenza:
  - Un linguaggio per Data Retrieval permette di trovare tutti gli oggetti che soddisfano esattamente le condizioni definite. Tali linguaggi (algebra relazionale, SQL) garantiscono una **risposta corretta e completa** e di manipolare la risposta.
  - Un sistema di IR, invece, potrebbe restituire anche oggetti non esatti (**risultati non pertinenti**); l'importante è che siano piccoli errori accettabili per l'utente.

Tecniche di IR: inverted index

- I sistemi di IR non operano sui documenti originali, ma su una vista logica degli stessi. Tradizionalmente i documenti di una collezione vengono rappresentati tramite un insieme di keyword.
- La capacità di memorizzazione dei moderni elaboratori permette talvolta di rappresentare un documento tramite l'intero insieme delle parole in esso contenute; si parla allora di **vista logica full text**.
- Per collezioni molto grandi **tale tecnica può essere inutilizzabile**; si utilizzano allora tecniche di modifica del testo per ridurre la dimensione della vista logica, che diventa un insieme di **index term**.
- Il modulo di gestione della collezione si occupa di creare gli opportuni indici, contenenti tali termini.

Tecniche di IR: inverted index

- Le tecniche di indicizzazione studiate per le basi di dati relazionali (ad es. B-Tree) non sono adatte per i sistemi di Information Retrieval.
- L'indice più utilizzato è l'indice invertito **(inverted index)**:
  - Viene memorizzato l'elenco dei termini contenuti nei documenti della collezione;
  - Per ogni termine, viene mantenuta una lista dei documenti nei quali tale termine compare.
- Tale tecnica è valida per query semplici (insiemi di termini); modifiche sono necessarie se si vogliono gestire altre tipologie di query (frasi, prossimità ecc.).

Tecniche di IR: inverted index

- Il numero di termini indicizzati viene ridotto utilizzando una serie di tecniche, tra cui:
- **Eliminazione delle stopword**: articoli, congiunzioni ecc.;
  - "Il cane di Luca" -> "cane Luca"
- **De-hyphenation**: divisione di parole con trattino;
  - "Sotto-colonnello" -> Sotto colonnello
- **Stemming**: riduzione alla radice grammaticale;
  - "Mangiano, mangiamo, mangiassi" -> "Mangiare"
- **Thesauri**: gestione dei sinonimi, omonimi, ipernonimi
  - "Casa" -> "Casa, magione, abitazione, …"
- L'utilizzo di tali tecniche non sempre migliora la qualità delle risposte ad una query.

Tecniche di IR: inverted index

- Avendo quindi non i documenti ma i loro inverted index, il processo di ricerca di informazioni viene riformulato:
  1. L'utente specifica un bisogno informativo.
  2. La query viene eventualmente trasformata…
  3. …per poi essere eseguita, utilizzando indici precedentemente costruiti, al fine di trovare documenti rilevanti;
  4. I documenti trovati vengono ordinati in base alla (presunta) rilevanza e ritornati in tale ordine all'utente;
  5. L'utente esamina i documenti ritornati ed eventualmente raffina la query, dando il via ad un nuovo ciclo.

Modelli per IR

- Formalmente un **modello di Information Retrieval** è una quadrupla (D, Q, F, R), dove:
  - D è un insieme di viste logiche dei documenti della collezione;
  - Q è un insieme di viste logiche (dette query) dei bisogni informativi dell'utente;
  - F è un sistema per modellare documenti, query e le relazioni tra loro;
  - R(qi, dj) è una funzione di ranking che associa un numero reale non negativo ad una query qj e un documento dj, definendo un ordinamento tra i documenti con riferimento alla query qj.
- I due modelli classici di Information Retrieval sono il **Modello booleano e quello vettoriale**.

Modelli per IR

- Il **modello booleano** è il modello più semplice; si basa sulla teoria degli insiemi e l'algebra booleana. Storicamente, è stato il primo ed il più utilizzato per decenni.
- I documenti vengono rappresentate come insiemi di termini.
- Le query vengono specificate come espressioni booleane, cioè come un elenco di termini connessi dagli operatori booleani AND, OR e NOT.
- La strategia di ricerca è basata su un criterio di decisione binario, senza alcuna nozione di grado di rilevanza: un documento è considerato rilevante oppure non rilevante.

Modelli per IR

- Il **modello vettoriale** è giustificato dall'osservazione che assegnare un giudizio binario ai documenti (1=rilevante, 0=non rilevante) è troppo limitativo.
- Nel modello vettoriale ad ogni termine nei documenti o nelle query viene assegnato un peso (un numero reale positivo).
- I documenti e le query sono rappresentati come vettori in uno spazio n-dimensionale (n = # di termini indicizzati).
- La ricerca viene svolta calcolando il grado di similarità tra il vettore che rappresenta la query e i vettori che rappresentano ogni singolo documento: i documenti con più alto grado di similarità con la query hanno più probabilità di essere rilevanti per l'utente.
- Il grado di similarità viene quantificato utilizzando una qualche misura, ad esempio il coseno dell'angolo tra i due vettori (che esprime la "vicinanza" fra i vettori).

Valutazione di un sistema di IR

- Come è possibile rispondere alla domanda "quale di questi due sistemi di IR funziona meglio"?
- Un sistema tradizionale di Data Retrieval può essere valutato oggettivamente, sulla base delle performance (velocità di indicizzazione, ricerca ecc.).
- In un sistema di IR tali valutazioni sono possibili, ma a causa della soggettività delle risposte alle query, le cose si complicano. Quello che si vorrebbe in qualche modo misurare è la "soddisfazione" dell'utente.
- Esistono delle misure standard per **valutare la bontà** delle risposte fornite da un sistema di IR: **precision e recall**

Web search

- L'IR è nata per gestire collezioni statiche e ben conosciute: testi di legge, enciclopedie ecc., ma quando la collezione di riferimento diventa il Web, le cose cambiano completamente:
- La collezione è dinamica, molto variabile nel tempo;
- Le dimensioni sono enormi;
- I documenti non sono sempre disponibili;
- Le query degli utenti sono ancora più imprecise e vaghe.
- Le tecniche utilizzate dai motori di ricerca possono quindi differire, ad esempio Pagerank (una cui variante è usata da Google) utilizza criteri diversi dalla IR standard

Structured IR

- L'IR nasce per dati non strutturati. Tuttavia, negli ultimi anni sta emergendo la necessità di applicare tecniche di IR anche a dati semistrutturati, in particolare a documenti XML; si parla allora allora di Structured Information Retrieval (SIR).
- Molte cose cambiano. Ad es. in IR la risposta ad una query è un elenco di documenti; in SIR, la risposta è un elenco di documenti XML? O frammenti di documenti XML? O singoli elementi?
- Ultimamente sono state avanzate proposte per estendere XQuery con capacità di ricerca full-text.

Schema

Dopo aver visto i confini della disciplina IR, vediamo nel dettaglio come opera:

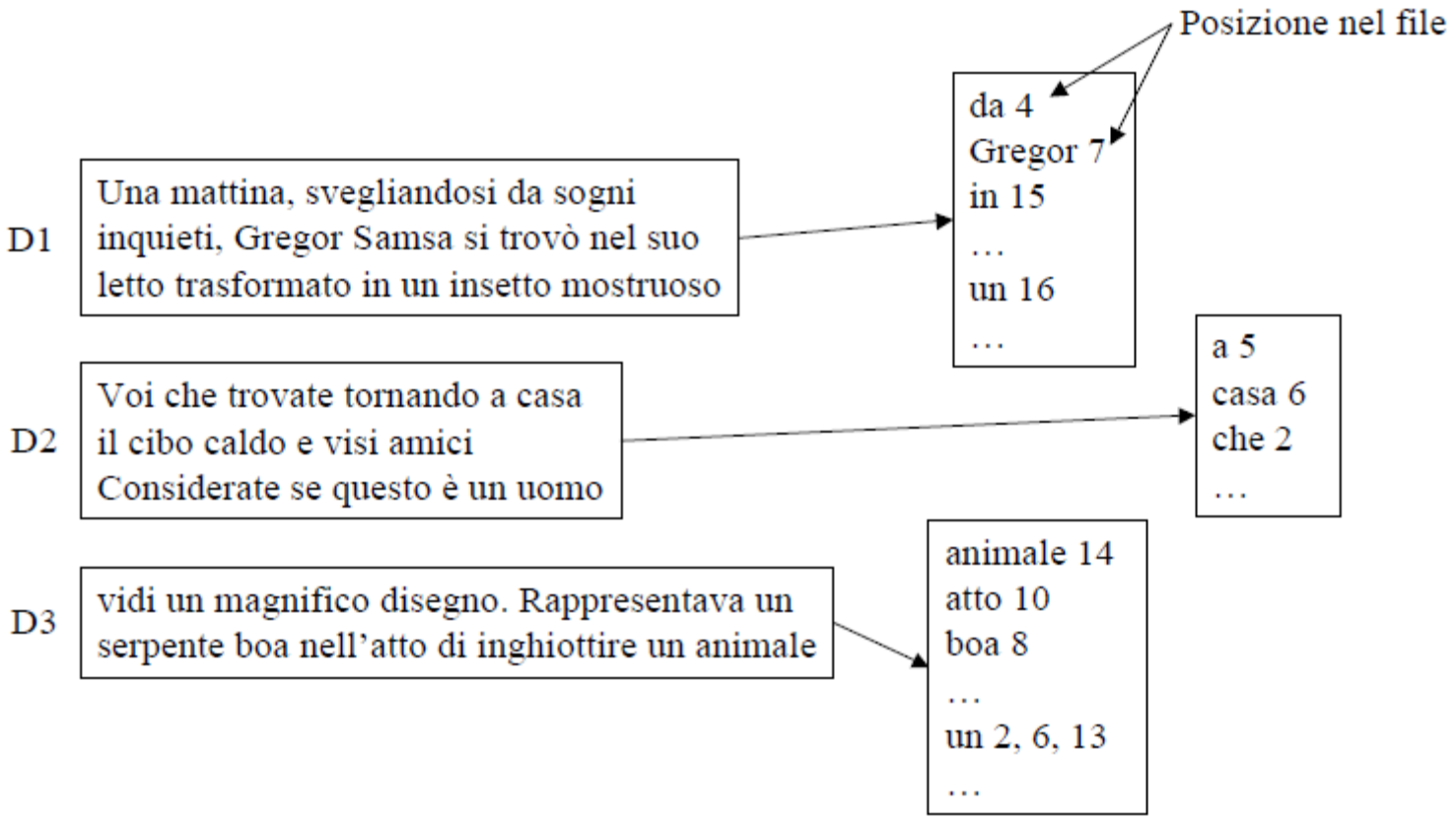
- Tecniche di IR: indicizzazione full text
- Modelli di IR: booleano e vettoriale
- Valutazione IR: precision e recall
- Advanced IR

Indicizzazione full-text: inverted indexes

- Nei sistemi relazionali gli indici permettono di recuperare efficientemente da una relazione i record contenenti uno o più valori di interesse. Analogamente, possiamo utilizzare indici per recuperare efficientemente documenti testuali di interesse. In questo caso, le interrogazioni che vogliamo supportare sono del tipo:
  - Ritornare i documenti che contengono l'insieme di keyword k1, …, kN.
  - Ritornare i documenti che contengono la sequenza di keyword k1, …, kN.
- Indici che supportano queste operazioni **si chiamano invertiti**, in quanto invece di rappresentare tutte le parole presenti in documento (indici) rappresentano tutti i documenti in cui appare una specifica parola.

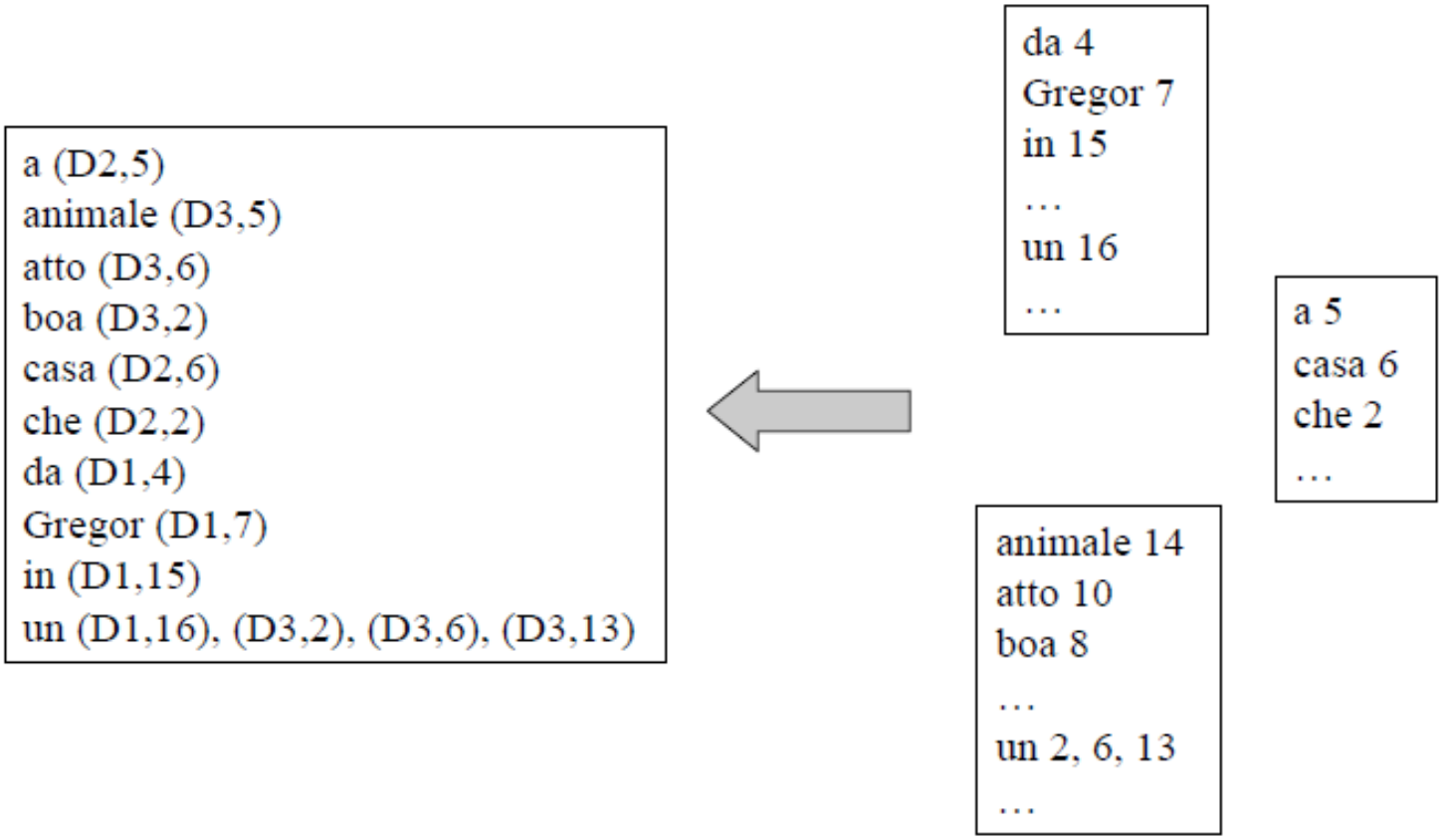
Indicizzazione full-text: inverted index

- Generazione degli indici: rilevamento della posizione dei terms



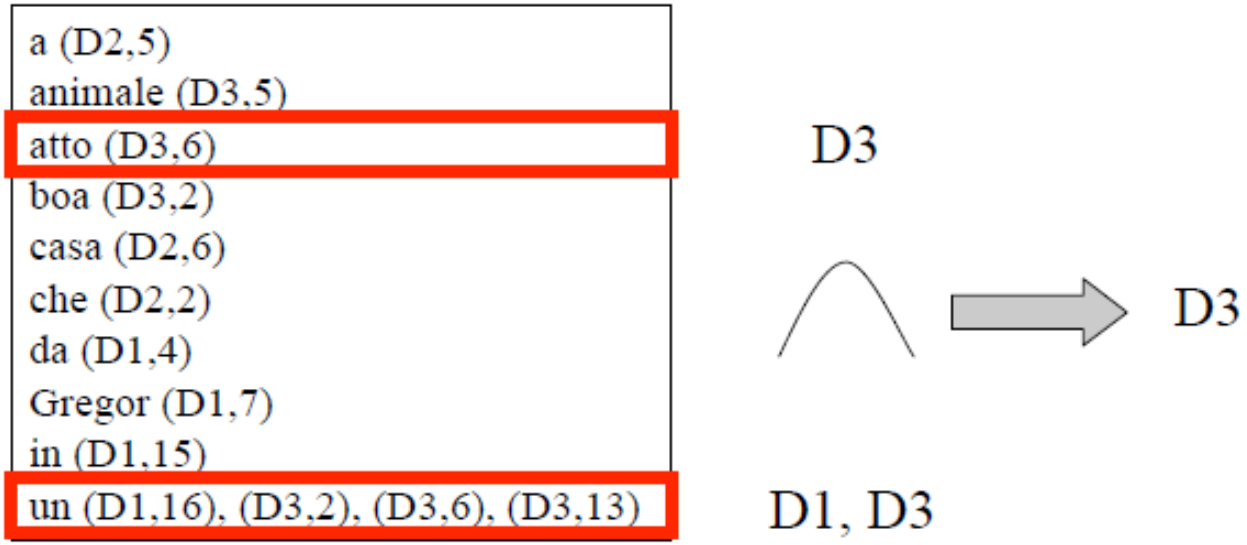
Indicizzazione full-text: inverted index

- Generazione degli indici: creazione dell'inverted index



Indicizzazione full-text: inverted index

- E' successivamente possibile effettuare delle interrogazioni:
  - Ritornare i documenti che contengono "un" e "atto".



D3

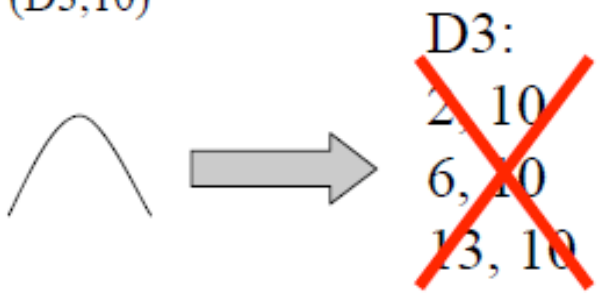
vidi **un** magnifico disegno. Rappresentava **un** serpente boa nell'**atto** di inghiottire **un** animale

Indicizzazione full-text: inverted index

- E' successivamente possibile effettuare delle interrogazioni:
  - Ritornare i documenti che contengono "un atto".

a (D2,5)
animale (D3,14)
atto (D3,10)
boa (D3,8)
casa (D2,6)
che (D2,2)
da (D1,4)
Gregor (D1,7)
in (D1,15)
un (D1,16), (D3,2), (D3,6), (D3,13)

(D3,10)



(D1,16), (D3,2), (D3,6), (D3,13)

Indicizzazione full-text: inverted index

- E' successivamente possibile effettuare delle interrogazioni:
  - Ritornare i documenti che contengono "un animale".

a (D2,5)
animale (D3,14)
atto (D3,10)
boa (D3,8)
casa (D2,6)
che (D2,2)
da (D1,4)
Gregor (D1,7)
in (D1,15)
un (D1,16), (D3,2), (D3,6), (D3,13)

(D3,14)

D3:
2, 14
6, 14
13, 14

(D1,16), (D3,2), (D3,6), (D3,13)

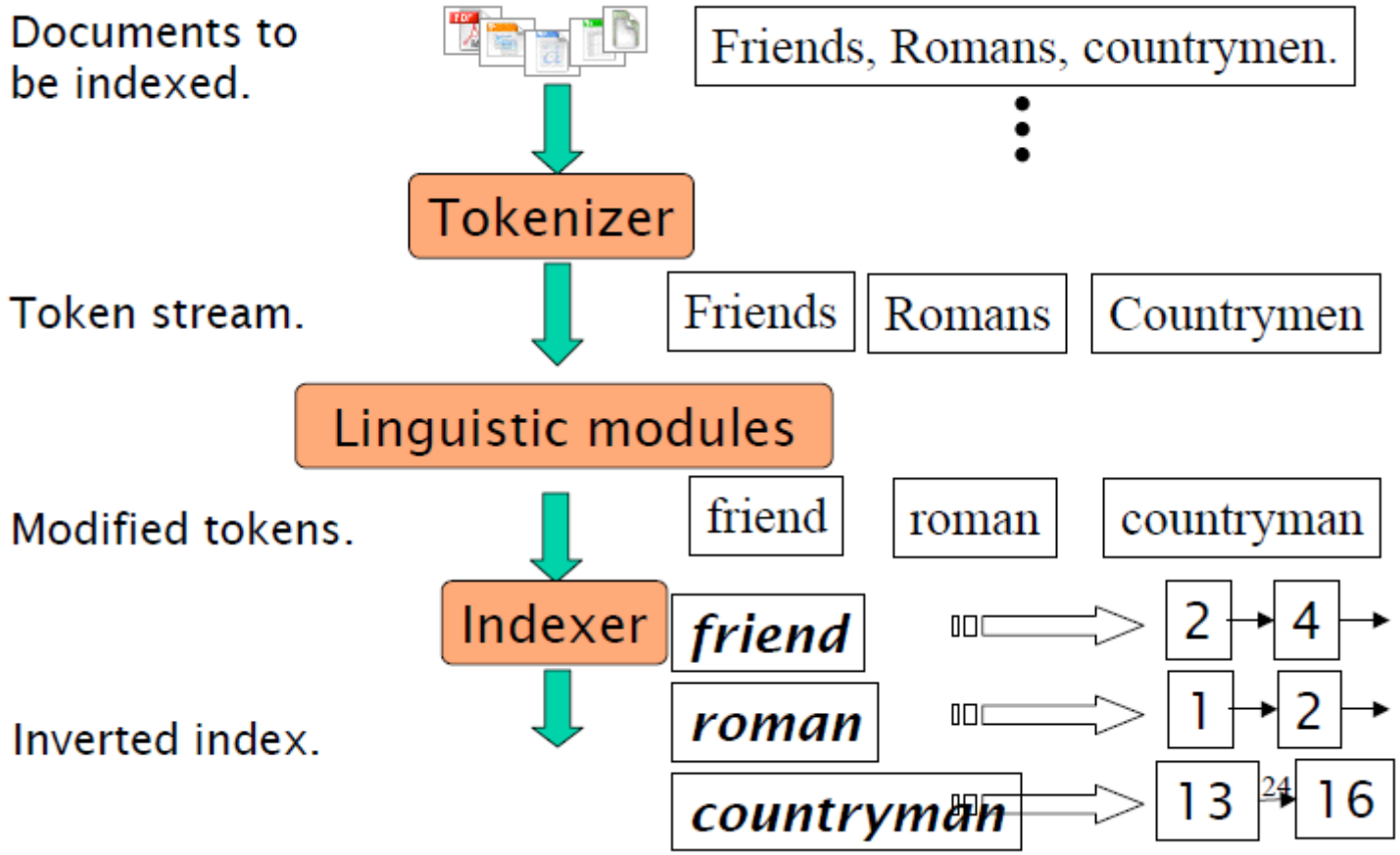
D3 | vidi un magnifico disegno. Rappresentava un serpente boa nell'atto di inghiottire **un animale**

Indicizzazione full-text: inverted index

- La **costruzione dell'inverted index è più complessa**, come visto infatti, gli indici invertiti lavorano su "termini", ma **cos'è un termine**?
  - Principi → Prìncipi? Princìpi?
  - Cordon Bleu → un termine o due?
  - Semi-structured → Semistructured? Semi-structured? Semi e Structured?
  - La, e, and, or, 192.168.0.1, 25/12/2004… → vanno indicizzati?
  - Auto e Automobile → lo stesso termine o due termini diversi?
  - Sono, Siamo, E' → lo stesso termine o tre termini diversi?
- Prima della fase di costruzione dell'indice, si devono quindi trasformare i documenti della collezione in un elenco di termini.

Indicizzazione full-text: inverted index

- Il processo di indicizzazione consta di diverse fasi:



Indicizzazione full-text: inverted index

- La prima fase è quella di tokenizzazione, in cui il tokenizer trasforma uno stream di testo ("Friends, Romans, Countrymen") in un elenco di token ("Friends", "Romans", "Countrymen") **candidati** a diventare entry dell'indice. **Tipiche trasformazioni** effettuate durante la tokenizzazione:
  - Eliminazione delle parole contenenti cifre;
  - Divisione in più parole dove è presente un trattino (de-hyphenation);
  - Trasformazione delle maiuscole in minuscole;
  - Eliminazione della punteggiatura.
- Ma è necessario **gestire alcune eccezioni**:
  - Trattini che sono parti integranti di una parola (es. B-49);
  - Parole che se scritte in maiuscolo assumono un diverso significato (es. MIT vs la parola tedesca mit);
  - Punteggiatura che è parte integrante di una parola (es. 510D.C.).

Indicizzazione full-text: inverted index

- Successivamente, entrano in azione i moduli linguistici, il cui scopo è prendere i token e validarli, operazione che dipende dalla lingua utilizzata (anche se alcuni criteri sono generali). Le operazioni effettuate dai moduli sono:
  - Eliminazione delle stopword
  - Stemming
  - Thesauri
  - Lemmatization

Indicizzazione full-text: inverted index

- **Eliminazione delle stopwords**: alcune parole sono più importanti di altre per comprendere il contenuto di un documento. Le parole che non hanno un significato proprio e sono eliminate sono le stopwords: articoli, congiunzioni, particelle pronominali, verbi frequenti ecc.
- Eliminare le stopwords attenua il rumore che disturba la ricerca di informazioni e riduce la dimensione dell'indice.
- Lo **stemming** riduce i termini alla loro "radice", rimuovendo prefissi e suffissi, ad esempio:
  - automate, automatic, automation → automat;
  - "for example compressed and compression are both accepted as equivalent to compress" → "for example compres and compres are both accept as equival to compres".
- Esistono vari algoritmi di stemming; il più comune per l'inglese è l'algoritmo di Porter, che opera trasformazioni come:
  - sses → ss (witnesses → witness);
  - s → Ø (cars → car);
  - tional → tion (national → nation).

Indicizzazione full-text: inverted index

- Con i **thesauri** si gestiscono i sinonimi tramite classi di equivalenza predefinite, ad esempio car = automobile. Due possibili tecniche:
  - **Espansione dell'indice**: se un documento contiene car, lo inseriamo nei posting sia di car che di automobile;
  - **Espansione della query**: se una query contiene car, cerchiamo anche i documenti contenenti automobile (preferibile).
  - L'utilizzo delle classi di equivalenza può però portare a risultati scorretti; ad es. puma e jaguar sono sinonimi, ma rischio di trovare informazioni relative alle automobili piuttosto che all'animale…
- La **lemmatization** riduce una parola alla sua radice grammaticale, ad esempio:
  - am, are, is → be;
  - car, cars, car's, cars' → car;
  - the boy's cars are different colors → the boy car be different color.

Indicizzazione full-text: inverted index

- Come visto, i moduli linguistici sono dipendenti dal linguaggio usato nel testo.
- **Possono crearsi problemi** se il testo contiene parole scritte in diversi linguaggi. Alcuni linguaggi creano inoltre problemi aggiuntivi: pensiamo al giapponese, cinese, arabo etc.
- Danno dei benefici in termini di precisione della ricerca e dimensione dell'indice.
- Ma trasformare il testo può rendere più difficile la ricerca all'utente: pensiamo alla ricerca "to be or not to be”.
- Per questo motivo non tutti sono concordi sull'opportunità di usarli (molti Web Search Engine non lo fanno).

Schema

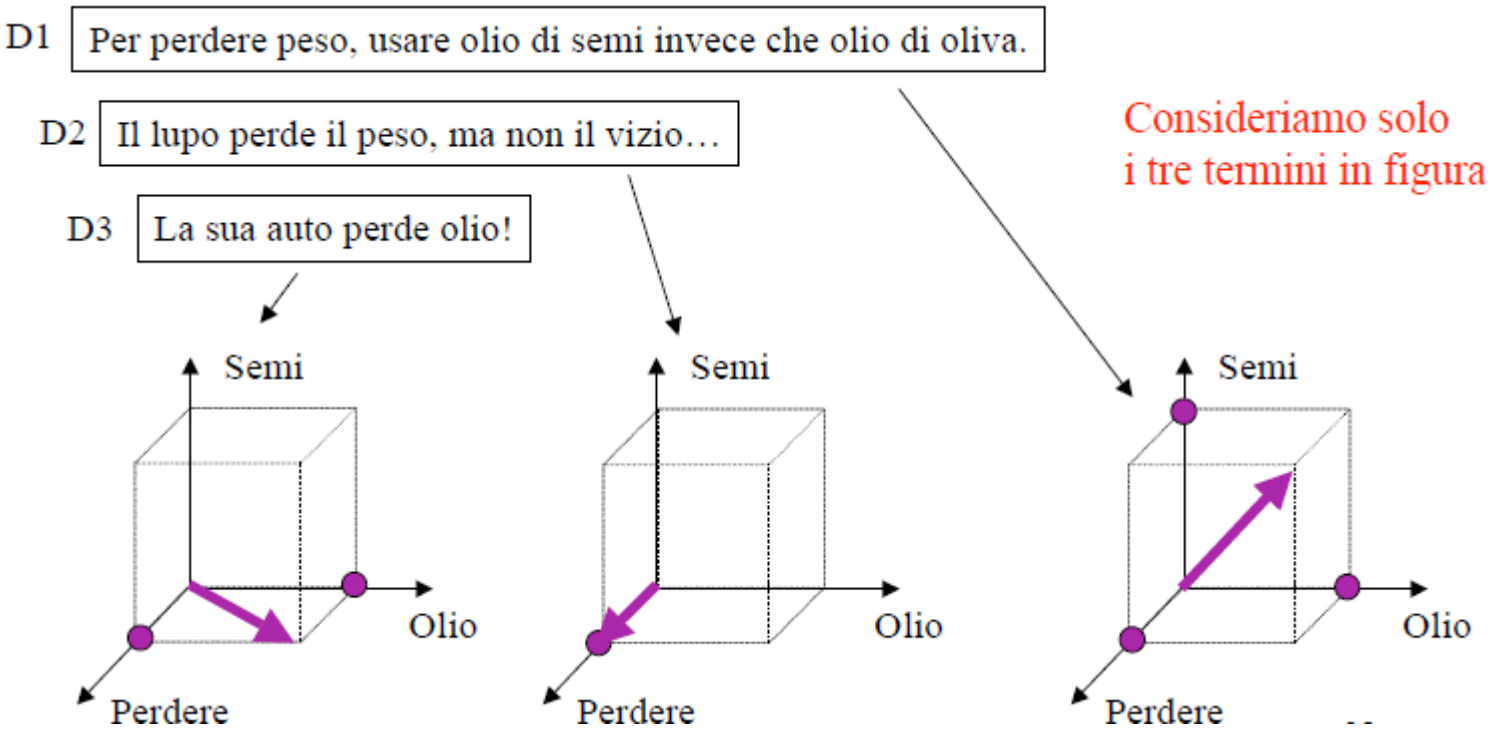
- Tecniche di IR: indicizzazione full text
- Modelli di IR: booleano e vettoriale
- Valutazione IR: precision e recall
- Advanced IR

Modelli di IR

- Nel modello booleano, un documento soddisfa le condizioni oppure no.
- Questo modello può essere ragionevole solo per degli utenti esperti che conoscano perfettamente la collezione di documenti e le proprie necessità.
- Una query può ritornare migliaia di risultati, ma la maggior parte degli utenti non vogliono scorrere migliaia di voci.
- Inoltre una eventuale riformulazione della query provoca il ricalcolo dell'intero risultato, con evidenti problemi prestazionali.
- Per questa serie di motivi, il modello booleano di fatto NON viene utilizzato, preferendo in sua vece quello vettoriale

Modelli di IR

- Ogni documento può essere visto come un vettore di valori in uno spazio vettoriale (n-dimensionale), i cui assi sono i termini, contenente i documenti. Le query possono essere viste come dei brevi documenti, e quindi anch'esse sono dei vettori appartenenti a questo spazio, esempio:



Modelli di IR

Modello vettoriale:

D1 | Per perdere peso, usare olio di semi invece che olio di oliva.

Semi	Perdere	Olio
[1	1	1]

D2 | Il lupo perde il peso, ma non il vizio…

Semi	Perdere	Olio
[0	1	0]

D3 | La sua auto perde olio!

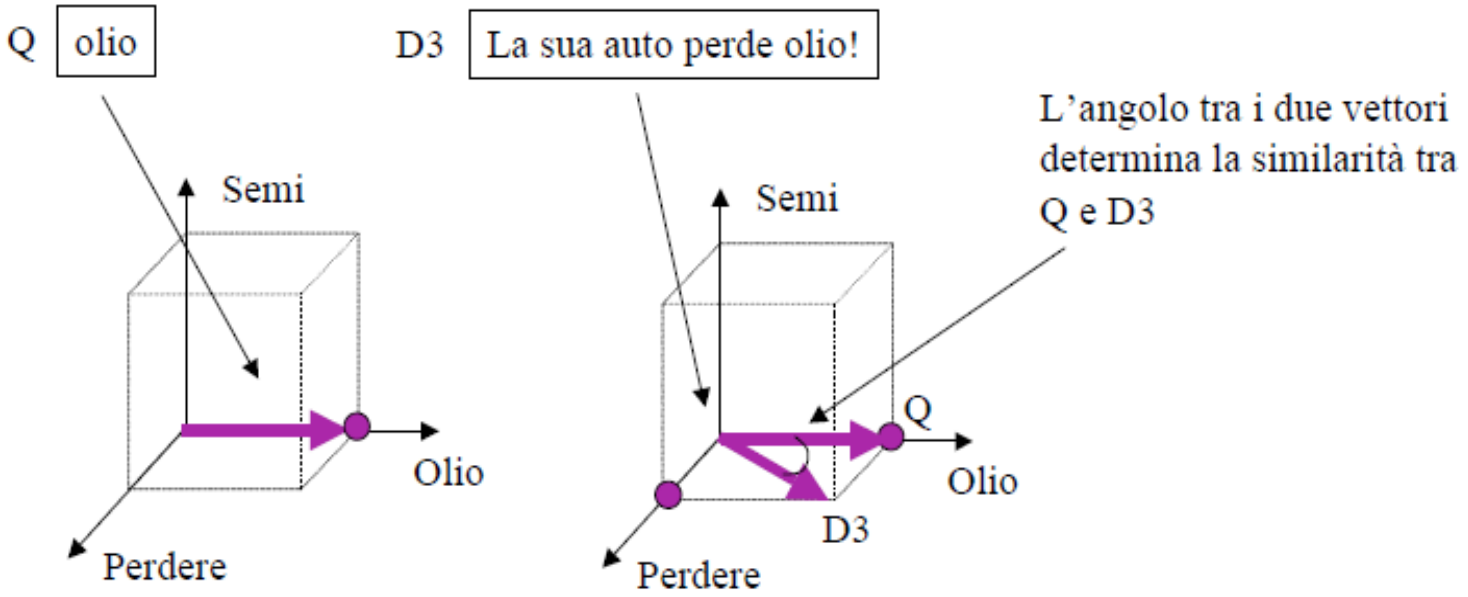
Semi	Perdere	Olio
[0	1	1]

Q | olio

Semi	Perdere	Olio
[0	0	1]

Modelli di IR

- Per esprimere il concetto di similitudine fra documenti e la query, quindi per rispondere ad una query, si introduce una **metrica**, ossia una **distanza fra vettori**, che banalmente potrebbe essere quella degli spazi lineari, ossia il **coseno dell'angolo** fra i vettori, che di fatto viene calcolato usando le coordinate. I documenti saranno poi ordinati (**ranking**) in base alla minore distanza dalla query



Modelli di IR

- Similarità fra i documenti dj e dk (uno dei quali solitamente è la query):

Prodotto scalare

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Prodotto della lunghezza dei due vettori

Modelli di IR

- Esempio:

	Semi	Perdere	Olio
D1	1	1	1
D2	0	1	0
D3	0	1	1
Q	0	0	1

$$\text{Sim}(D1, Q) = \frac{1*0 + 1*0 + 1*1}{\sqrt{3} * \sqrt{1}} = .577$$

$$\text{Sim}(D2, Q) = \frac{0*0 + 1*0 + 0*1}{\sqrt{1} * \sqrt{2}} = 0$$

$$\text{Sim}(D3, Q) = \frac{0*0 + 1*0 + 1*1}{\sqrt{2} * \sqrt{1}} = .707$$

Modelli di IR

- I problemi principali di questo approccio sono:
  - Olio appare due volte nel primo documento, il che può indicare una maggiore importanza della parola.
  - Perdere appare in tutti i documenti, per cui non serve a discriminare tra più o meno rilevanti.
- Invece di segnalare la presenza o meno di un termine in un documento, vogliamo dunque assegnarvi un **peso**. Un approccio tipico consiste nell'utilizzare pesi ottenuti dal prodotto tra la **term frequency** (tf, ovvero, quante volte o in che percentuale un termine appare nel documento) e **l'inverse document frequency** (idf, ovvero, quanto è rara l'occorrenza di un termine).

Modelli di IR

- Per **Document Frequency** si intende il numero di documenti che contengono un certo termine.
- L'inverso del Document Frequency (**Inverse Document Frequency**, idf), cioè la rarità di un termine all'interno della collezione, è una buona misura della significatività di un termine.
- Solitamente viene usata la seguente formula: $idf_i = \log (N/df_i)$
- Dove N = numero totale di documenti della collezione e dfi = numero di documenti che contengono il termine i.
- Ad ogni termine della query viene assegnato un peso in base ad una misura combinata di tf e idf (tf / idf):
- $w_{i,d} = tf_{i,d} \times \log (N / df_i)$

Modelli di IR

- Esempio:

D1 | Per **perdere** peso, usare **olio** di **semi** invece che **olio** di oliva.

Semi	Perdere	Olio
[1 * log(3/1)	1 * log(3/3)	2 * log(3/2)]

Perdere appare 1 volta in D1

Perdere appare in 3 documenti su 3

Olio appare 2 volte in D1

Olio appare in 2 documenti su 3

D2 | Il lupo **perde** il peso, ma non il vizio…

D3 | La sua auto **perde olio**!

Modelli di IR

- Esempio:

	Semi	Perdere	Olio
D1	.477	0	.252
D2	0	0	0
D3	0	0	.176
Q	0	0	.176

$$\text{Sim}(D1, Q) = \frac{.447*0 + 0*0 + .252*.176}{.512 * .176} = .492$$

$$\text{Sim}(D2, Q) = \frac{0*0 + 0*0 + 0*.176}{0 * .176} = 0$$

$$\text{Sim}(D3, Q) = \frac{0*0 + 0*0 + .176*.176}{.176 * .176} = 1$$

Modelli di IR

- Problemi dell'approccio vettoriale:
- La lunghezza dei documenti incide sul calcolo della rilevanza.
- Il numero di documenti incide sul calcolo della rilevanza.
- Non viene considerato l'ordine dei termini.

Schema

- Tecniche di IR: indicizzazione full text
- Modelli di IR: booleano e vettoriale
- Valutazione IR: precision e recall
- Advanced IR

Valutazione IR

- Un sistema tradizionale di Data Retrieval può essere valutato utilizzando **svariate misure**:
  - Velocità di indicizzazione (numero di documenti indicizzati all'ora);
  - Velocità di ricerca (in funzione della dimensione dell'indice);
  - Espressività del linguaggio di interrogazione.
- Tutte queste proprietà (performance evaluation) sono misurabili.
- Ma la vera e più importante misura delle prestazioni di un motore di IR è un'altra: la "soddisfazione" dell'utente (retrieval performance evaluation), visto il meccanismo ranking based
- **Come misurare la soddisfazione di un utente**? La velocità di ricerca è sicuramente un fattore importante, ma una risposta velocissima ma inutile non renderà felice l'utente!

Valutazione IR

- Misurare il grado di soddisfazione di un utente tuttavia **non è cosa facile**, la scelta più valida infatti dipende dal tipo di utente e di applicazione:
- **Motore di ricerca**: se un utente è soddisfatto delle prestazioni di un motore di ricerca tornerà ad utilizzarlo, quindi potremmo misurare la percentuale di utenti che "tornano";
- **Sito di e-commerce**: dal punto di vista dell'utilizzatore del sito, il motore è valido se il tempo necessario per effettuare un acquisto è basso; dal punto di vista del proprietario del sito, il motore è buono se un'alta percentuale di ricerche si concludono con un acquisto;
- **Azienda**: una valida misura può essere il tempo risparmiato dai dipendenti nella ricerca di informazioni.

Valutazione IR

- In generale allora il modo migliore per valutare un motore di IR è considerare la **rilevanza dei risultati**. Occorre definire una metodologia ed abbiamo bisogno di una serie di strumenti:
- Una **collezione di documenti di test**; esistono diverse collezioni, tra cui ricordiamo la collezione TREC, sviluppata da NIST (National Institute of Standards and Technology): circa 700K documenti, dimensione 2GB
- **Un elenco di esempi** di richieste di informazioni (query), solitamente definite informalmente, in linguaggio naturale (si parla perciò più precisamente di retrieval task), e da esperti del settore
- **Una valutazione di rilevanza**, cioè un giudizio rilevante / non rilevante per ogni coppia query / documento, definita da esperti del settore
- Data una strategia di IR, la misura della valutazione quantifica la similarità tra l'insieme dei documenti ritornati e l'insieme dei documenti classificati come rilevanti.

Valutazione IR

- La metodologia di valutazione più utilizzata si basa su due misure:
- **Precision**: percentuale di documenti ritornati (in risposta ad una query) che sono rilevanti;
- **Recall**: percentuale di documenti rilevanti che sono ritornati.

$$\text{Precision} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Ritornati}\}|}$$

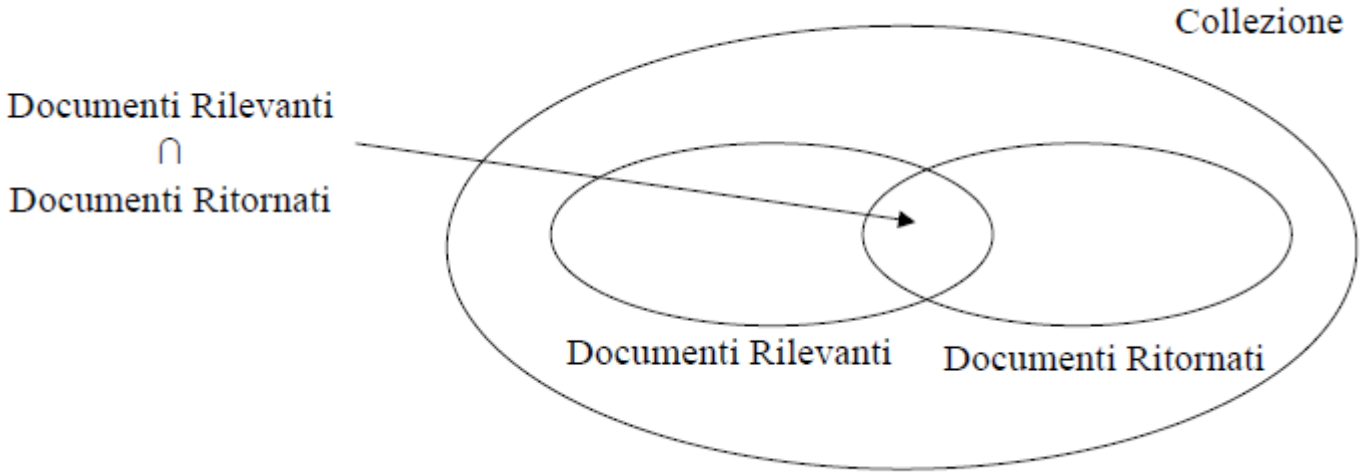
$$\text{Recall} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Rilevanti}\}|}$$

Valutazione IR

- Precision e recall:

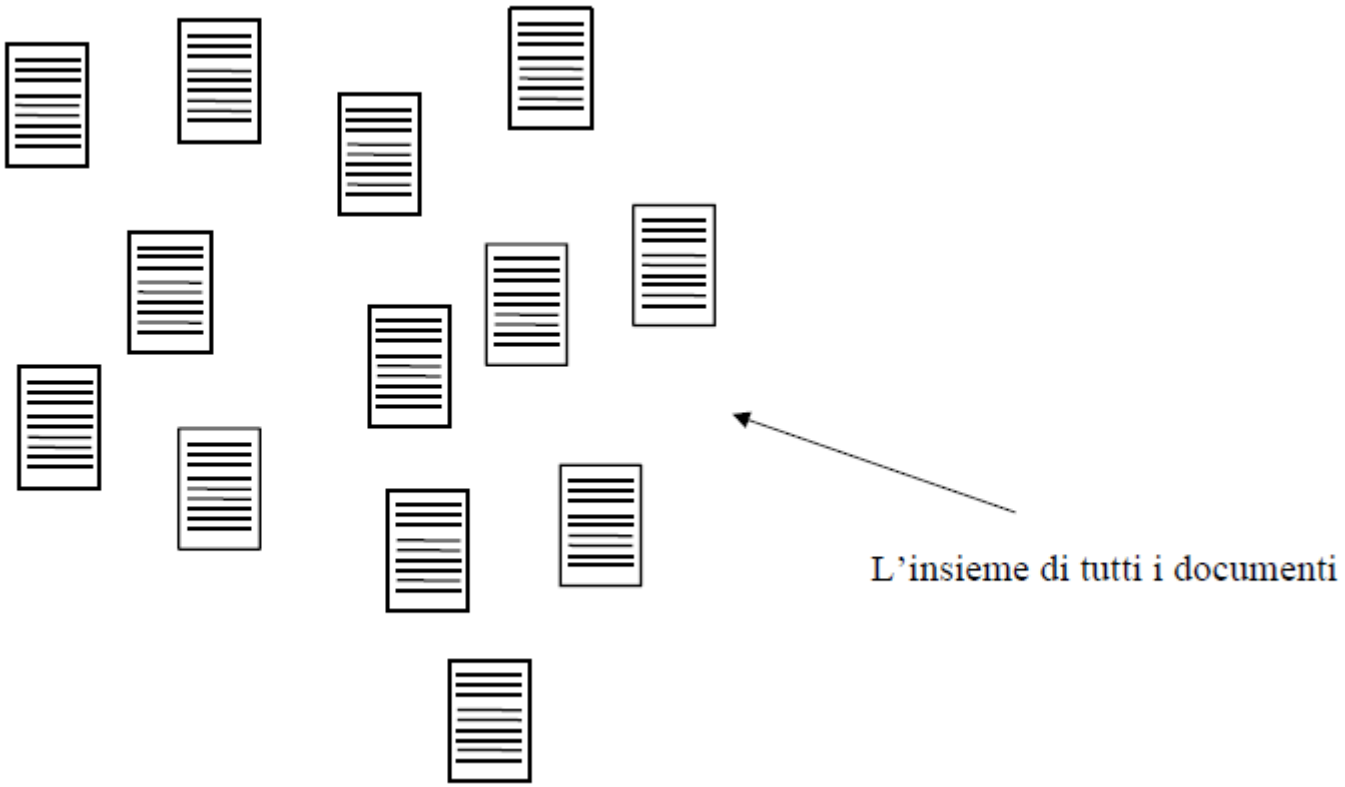
$$\text{Precision} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Ritornati}\}|}$$

$$\text{Recall} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Rilevanti}\}|}$$



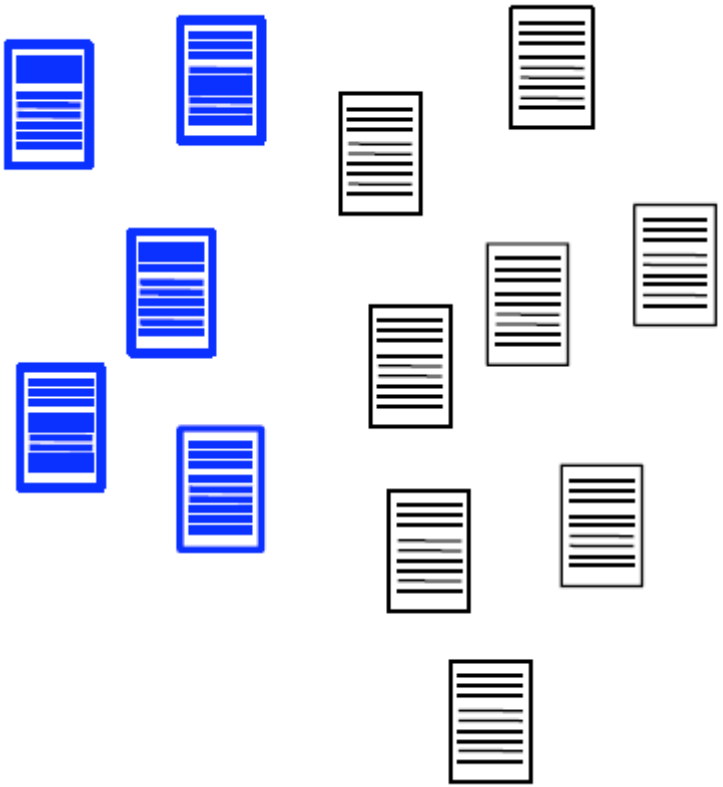
Valutazione IR

- Precision e recall:



Valutazione IR

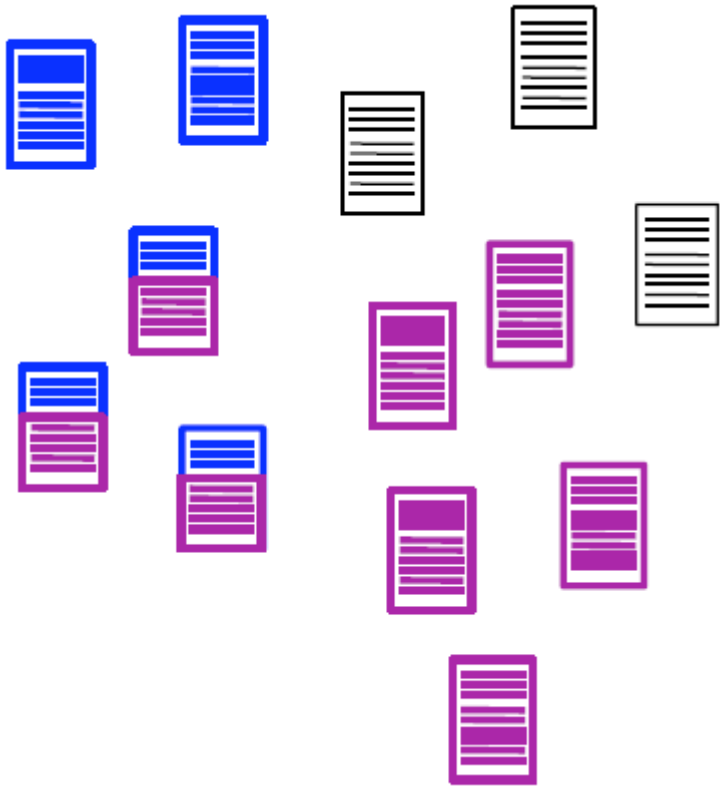
- Precision e recall:



DOCUMENTI RILEVANTI

Valutazione IR

- Precision e recall:



DOCUMENTI RILEVANTI

RISULTATO DELLA QUERY
(DOCUMENTI RITORNATI)

Valutazione IR

- Precision e recall:

DOCUMENTI RILEVANTI

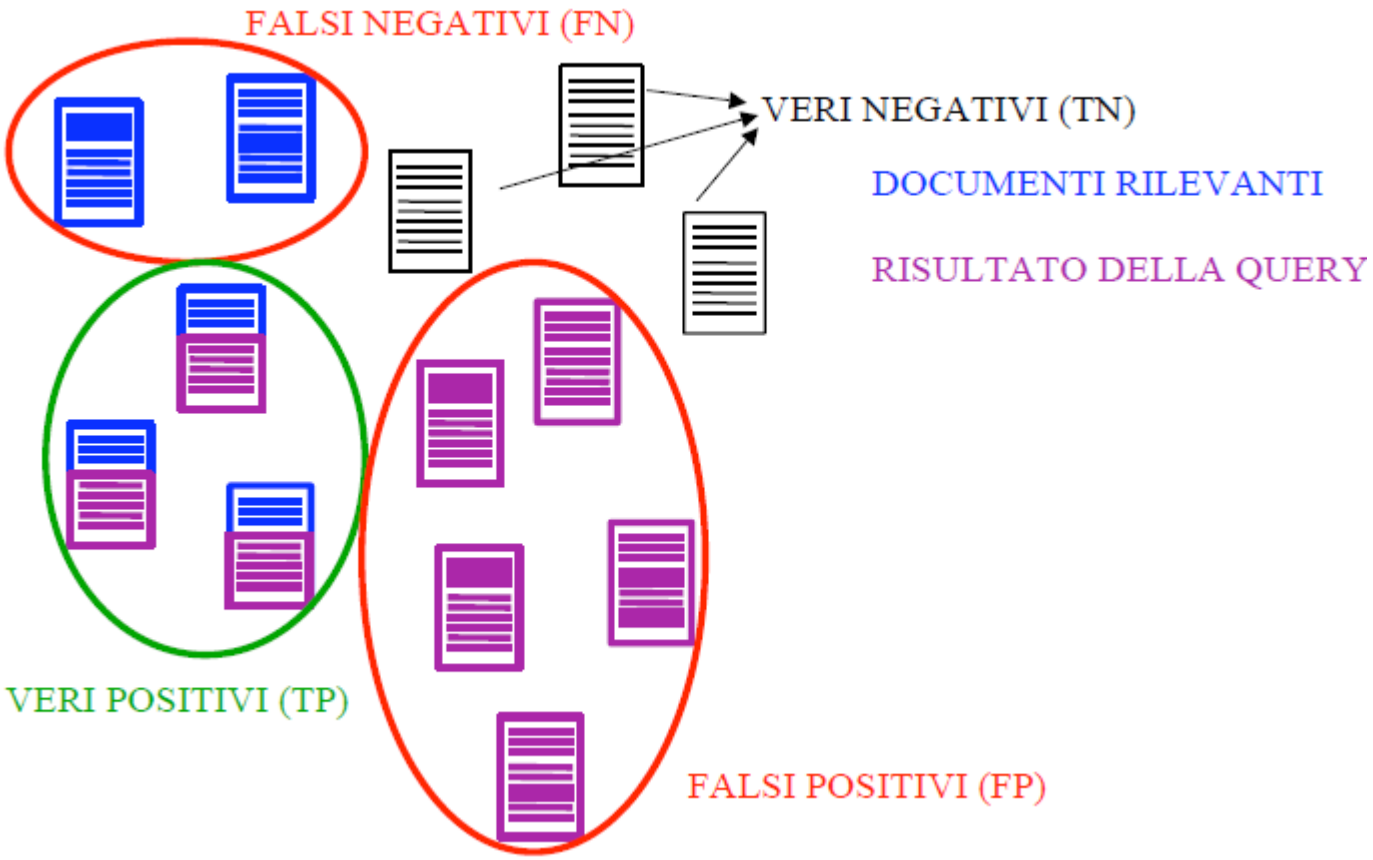
RISULTATO DELLA QUERY
(DOCUMENTI RITORNATI)

$$\text{Precision} = \frac{|\{\text{Rilevanti}\} \cap \{\text{Ritornati}\}|}{|\{\text{Ritornati}\}|} = 3/8$$

$$\text{Recall} = \frac{|\{\text{Rilevanti}\} \cap \{\text{Ritornati}\}|}{|\{\text{Rilevanti}\}|} = 3/5$$

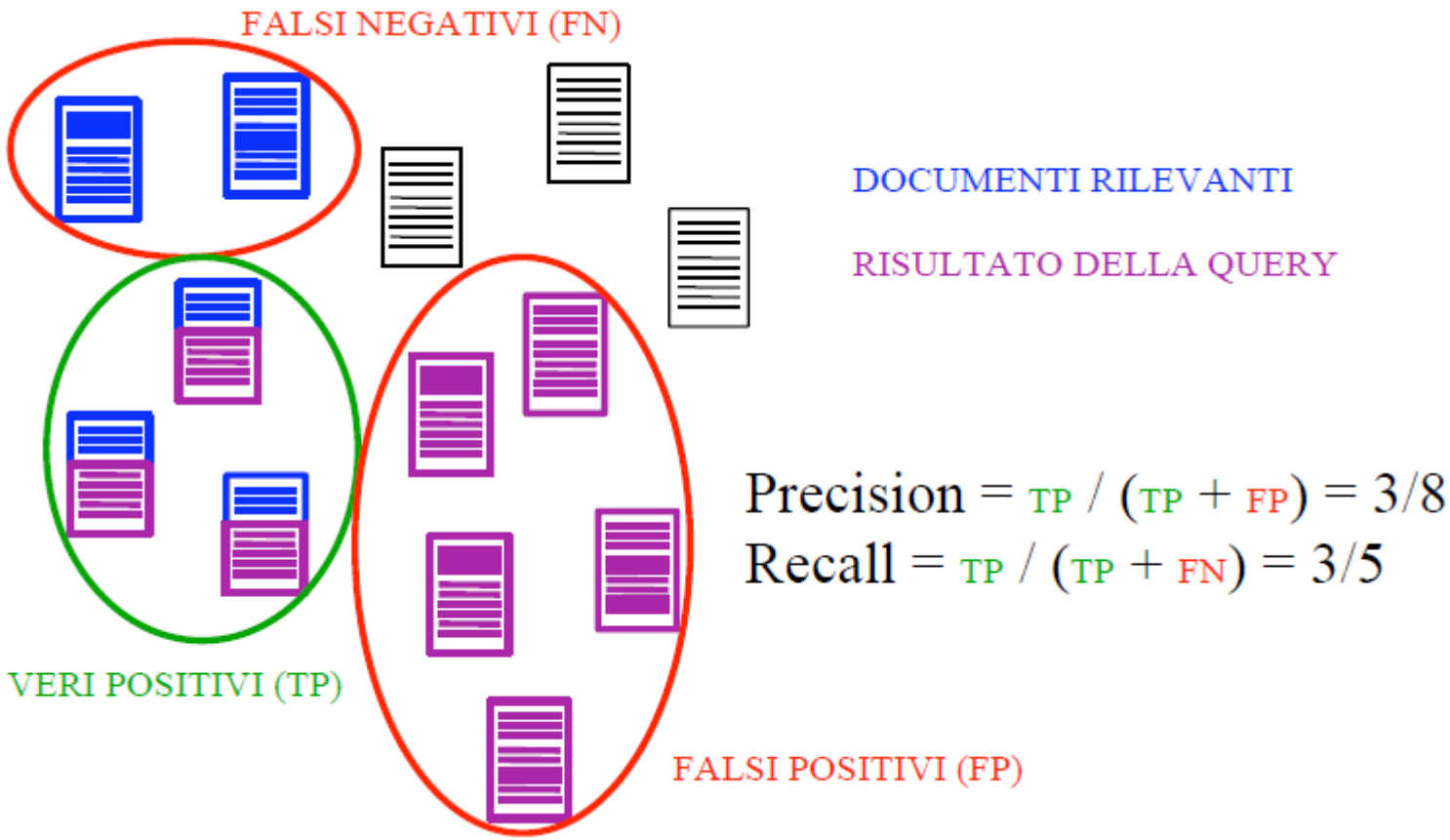
Valutazione IR

- Precision e recall:



Valutazione IR

- Precision e recall



$$\text{Precision} = TP / (TP + FP) = 3/8$$
$$\text{Recall} = TP / (TP + FN) = 3/5$$

Valutazione IR

- Precision e recall: riassumendo

Precision $P = tp / (tp + fp)$.

Recall $R = tp / (tp + fn)$.

	Rilevanti	Non rilevanti
Ritornati	*tp* (True Positive)	*fp* (False Positive)
Non ritornati	*fn* (False Negative)	*tn* (True Negative)

O, EQUIVALENTEMENTE,

$$\text{Precision} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Ritornati}\}|}$$

$$\text{Recall} = \frac{|\{\text{Documenti Rilevanti}\} \cap \{\text{Documenti Ritornati}\}|}{|\{\text{Documenti Rilevanti}\}|}$$

Valutazione IR

- **Perché utilizzare due misure?**
- Se considerassimo solo la misura Recall, un motore che (in risposta a qualsiasi query) ritorna tutti i documenti della collection sarebbe considerato perfetto (Recall = 1).
- Se considerassimo solo la misura Precision, un motore che ritorna solo un documento (che sicuramente viene considerato rilevante) sarebbe considerato perfetto (Precision = 1).
- Precision e Recall, considerate come due funzioni del numero di documenti ritornati, hanno un andamento opposto:
- Recall è non decrescente;
- Precision è solitamente decrescente.
- Usando questi comportamenti, nel valutare le prestazioni di un motore di ricerca che restituisce un elenco di documenti ordinati per similarità con la query, possiamo valutare Precision e Recall a vari livelli di Recall, cioè variando il numero di documenti ritornati. Otteniamo così delle curve di Precision e Recall

Valutazione IR

- **Combinando precision e recall**, si ha F, che è una misura combinata che bilancia l'importanza di Precision e Recall.
- Solitamente viene usata la misura bilanciata F1 (cioè con β = 1 o α = ½). In questa misura combinata si assume che l'utente bilanci l'importanza di Precision e Recall.

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Valutazione IR

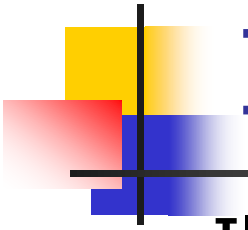
- **In conclusione**, l'utilizzo di Precision e Recall per la valutazione di un motore di IR pone **alcuni problemi**:
- I documenti della collezione devono essere valutati manualmente da persone esperte: non sempre il giudizio è completamente veritiero;
- La valutazione dei documenti è binaria (rilevante / non rilevante): non sempre è facile catalogare così nettamente un documento;
- Le misure sono pesantemente influenzate dal dominio di applicazione, cioè dalla collezione e dalle query: un motore di IR potrebbe avere delle ottime prestazioni in un determinato dominio ma non in un'altro.

Schema

- Tecniche di IR: indicizzazione full text
- Modelli di IR: booleano e vettoriale
- Valutazione IR: precision e recall
- Advanced IR

Advanced IR

- Per incrementare l'efficiacia dell'IR, diverse tecniche sono utilizzate:
  - Ranking probabilistico
  - Latent Semantic Indexing
  - Relevance FeedBack e Query Expansion



IR probabilistico

- Il modello probabilistico: Il principio di pesatura probabilistico, o *probability ranking principle*
- Metodi di ranking:
  - Binary Independence Model
  - Bayesian networks
- **L'idea chiave è di classificare i documenti in ordine di probabilità di rilevanza rispetto all'informazione richiesta:**

P(rilevante|documento$_i$, query)

Probability Ranking Principle

•Sia *d* un documento della collezione.
•Sia *R* la **rilevanza** di un documento rispetto ad una (specifica) query (R=1) e sia *NR* la **non-rilevanza** (R=0).

Si vuole stimare p(*R*|*d*,*q*) - la probablità che *d* sia **rilevante, data la query q.** In base al teorema di Bayes:

$$p(R \mid d,q) = \frac{p(d \mid R,q)p(R \mid q)}{p(d \mid q)}$$

$$p(NR \mid d,q) = \frac{p(d \mid NR,q)p(NR \mid q)}{p(d \mid q)}$$

p(*R*|*q*),p(*NR*|*q*) - prob. a priori di recuperare un documento (non) rilevante

$$p(R \mid d,q) + p(NR \mid d,q) = 1$$

p(*d*|*R*,*q*), p(*d*|*NR*,*q*) - probabilità che, se si trova un documento rilevante (non-rilevante), questo sia *d*.

Probability Ranking Principle

Il **teorema di Bayes** si usa quando un evento B può verificarsi sotto diverse condizioni sulle quali si possono fare *n ipotesi.* Se si conosce la probabilità delle ipotesi nonché le probabilità condizionate, si potrà verificare se le ipotesi iniziali erano corrette o se devono essere modificate.

Considerando un insieme di alternative $A_1, A_2, \ldots A_n$ (partizione dello spazio degli eventi) si trova la seguente espressione per la probabilità condizionata:

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{P(E)} = \frac{P(E|A_i)P(A_i)}{\sum_{j=1}^{n} P(E|A_j)P(A_j)}$$

Dove:

- P(A) è la probabilità a priori o probabilità marginale di A. "A priori" significa che non tiene conto di nessuna informazione riguardo E.
- P(A|E) è la probabilità condizionata di A, noto E. Viene anche chiamata probabilità a posteriori, visto che è derivata o dipende dallo specifico valore di E.
- P(E|A) è la probabilità condizionata di E, noto A.
- P(E) è la probabilità a priori di E, e funge da costante di normalizzazione.

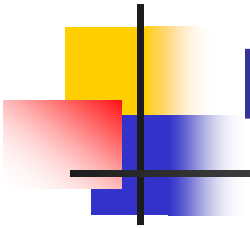
Intuitivamente, il teorema descrive il modo in cui le opinioni nell'osservare A siano arricchite dall'aver osservato l'evento E.

Probability Ranking Principle (PRP)

- Bayes' Optimal Decision Rule
  - d è rilevante iff p(R|d,q) > p(NR|d,q)

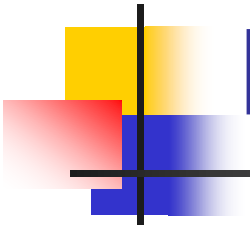
Modellando il processo di retrieval in termini probabilistici, l'occorrenza di una query, la rilevanza o non rilevanza di un documento, l'occorrenza di un termine in un documento sono tutti eventi **aleatori**

- Come si calcolano le probabilità condizionate?
  - Si usano "stimatori"
  - Il modello più semplice è il **Binary Independence Retrieval** (BIR)
  - Alternativamente, sono usate le **Reti Bayesiane**



Probability Ranking Principle (PRP)

- Si modella un problema in termini probabilistici (es: la rilevanza di un documento rispetto ad una query è stimata dalla P(R|d,q))
- Poiché in generale è difficile stimare una certo modello probabilistico, si effettuano una serie di passaggi (ad es. invertire variabile aleatoria condizionante e condizionata con Bayes) e semplificazioni (ad es. assumere l'indipendenza statistica di certe variabili) al fine di rappresentare il modello probabilistico iniziale in termini di probabilità più facili da stimare su un campione.



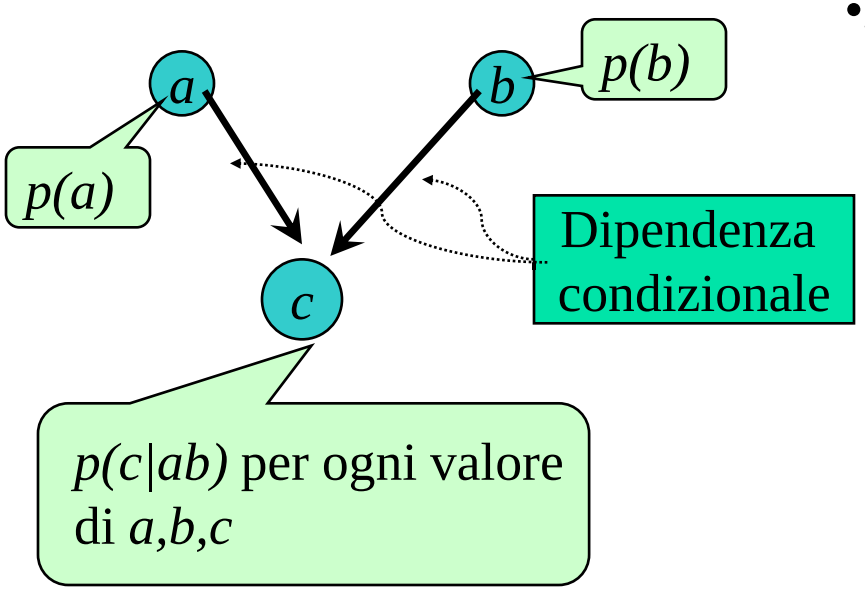
Bayesian Networks

Cosa è una Bayesian network?

- Un grafo aciclico diretto DAG
- Nodi: Eventi, variabili aleatorie, o variabili che possono assumere valori; per semplicità, nel modell BN-IR, tali valori si assumono booleani
- Archi: Modellano una dipendenza diretta fra nodi

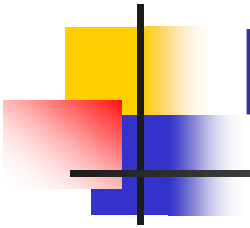
Bayesian Networks

a,*b*,*c* - nodi



$$P(c) = P(c/a)P(a) + P(c/b)P(b)$$

- Le reti Bayesiane modellano la dipendenza fra eventi
- Inference in Bayesian Nets:
  - note le probabilità a priori per le radici del grafo e le probabilità condizionate (archi) si può calcolare la probabilità a priori *di ogni evento condizionato.*
  - Se sono noti i valori di verità di alcuni nodi (ad esempio, l'osservazione dell'evento b e di ¬a) si possono ricalcolare le probabilità dei nodi



Bayesian Networks

- Obiettivo
  - Data una richiesta di informazione da parte di un utente (**evidenza**) stima la probabilità che un documento soddisfi la richiesta (**inferenza**)
- Modello di Retrieval
  - Modella i documenti come una rete *document network*
  - Modella il bisogno informativo come una *query network*

Belief Network Model: un modello di *ranking* basato su Reti Bayesiane

Definizioni:

K=$\{k_1, k_2, ..., k_t\}$ *spazio di campionamento* (o **spazio dei concetti**)

u $\subset$ **K** un subset di **K** (un **concetto**)

k_i un termine indice (***concetto elementare***)

k=$(k_1, k_2, ..., k_n)$ $n \leq t$ un vettore associato ad ogni concetto u tale che $g_i(\mathbf{k})=1 \Leftrightarrow k_i \in u$ (pesi unitari)

k_i una variabile aleatoria *binaria* (cioè $ki \in \{0,1\}$) associata al termine indice k_i , t.c. $k_i = 1 \Leftrightarrow g_i(\mathbf{k})=1 \Leftrightarrow k_i \in u$

Belief Network Model

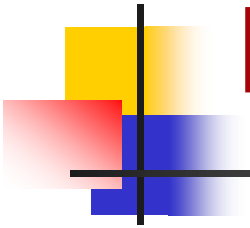
Il *ranking* di un documento d_j rispetto ad una query q è interpretato come una *relazione di corrispondenza fra concetti*, e riflette il **grado di copertura che il concetto d_j fornisce al concetto q.**

Documenti e query sono trattati nello stesso modo, cioè sono entrambi concetti nello spazio K.

Assunzione:

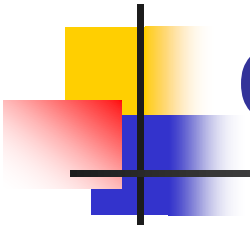
$P(d_j|q)$ viene considerato come il *rank* del documento d_j rispetto alla query q.

http://portal.acm.org/citation.cfm?id=243272 (Ribeiro and Munz, 1996: "A belief network model for IR")



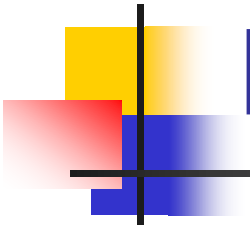
Belief Network Model

- Vantaggi del Belief Network Model:
- Per calcolare il *rank* di un documento, considera solo gli stati della rete in cui i nodi attivi sono quelli che compaiono nella query, quindi il costo è lineare nel numero dei documenti della collezione
- E' una variante moderna dei metodi di ragionamento probabilistico, che consente una combinazione di distinte sorgenti di evidenza. I modelli più avanzati consentono di incorporare nel modello evidenze derivate da sessioni precedenti, e feedback dell'utente.



Conclusioni sul Ranking probabilistico

- I modelli probabilistici rappresentano il problema del retrieval mediante probabilità condizionate (es. P(R/q,d)).
- Alcuni modelli consento di "rilassare" l'ipotesi di indipendenza fra termini
- Occorre stimare le probabilità condizionate fra termini (in genere bigrammi o trigrammi P(ti/tj) o P(ti/tj,tk)
- Fra i metodi per determinare correlazioni fra termini c'è il Latent Semantic Indexing, che è un metodo algebrico per stimare la similarità fra documenti, e fra documenti e query

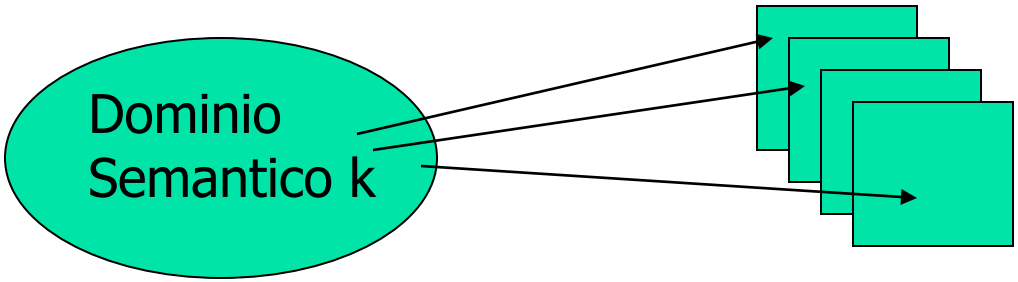


Latent Semantic Indexing

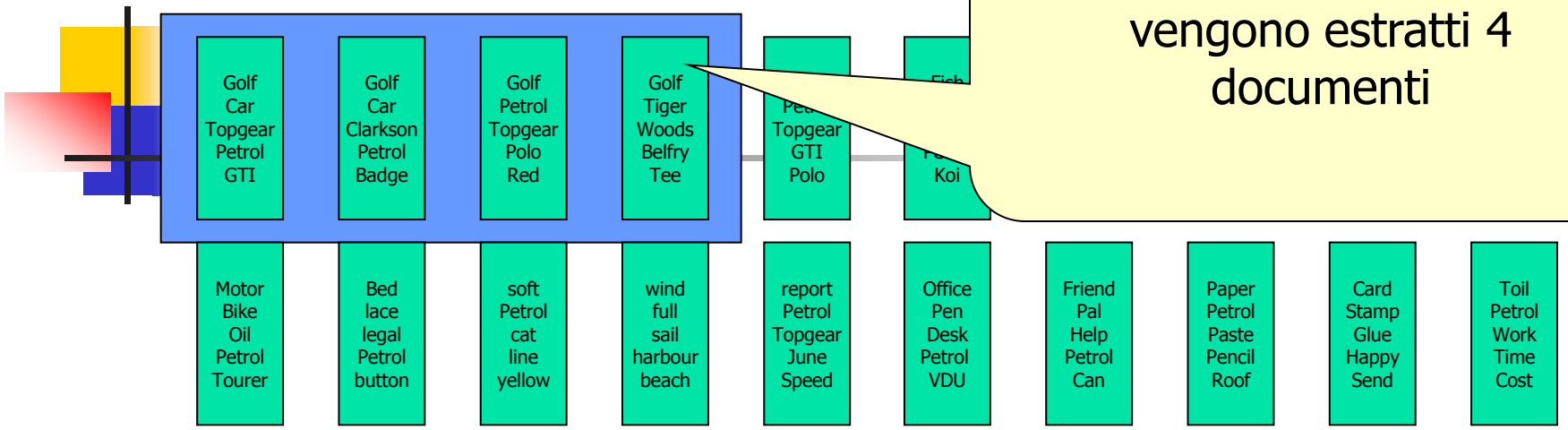
- I metodi di ranking tradizionali calcolano l'attinenza di un documento ad una query sulla base della presenza o meno di parole contenute nella query: un termine o è presente o non lo è
- Nel LSI la ricerca avviene per concetti: ma un concetto non è l'astrazione-generalizzazione di un termine (es: *golf→vestiario*) bensì un insieme di termini correlati (golf, maglia, vestito) detti co-occorrenze o **dominio semantico**

Latent Semantic Indexing

- Data una collezione di documenti, LSI è in grado di rilevare che alcune n-uple di termini co-occorrono frequentemente (es: gerarchia, ordinamento e classificazione)
- Se viene fatta una ricerca con gerarchia, **ordinamento** vengono "automaticamente" recuperati documenti che contengono anche (e eventualmente solo!) classificazione

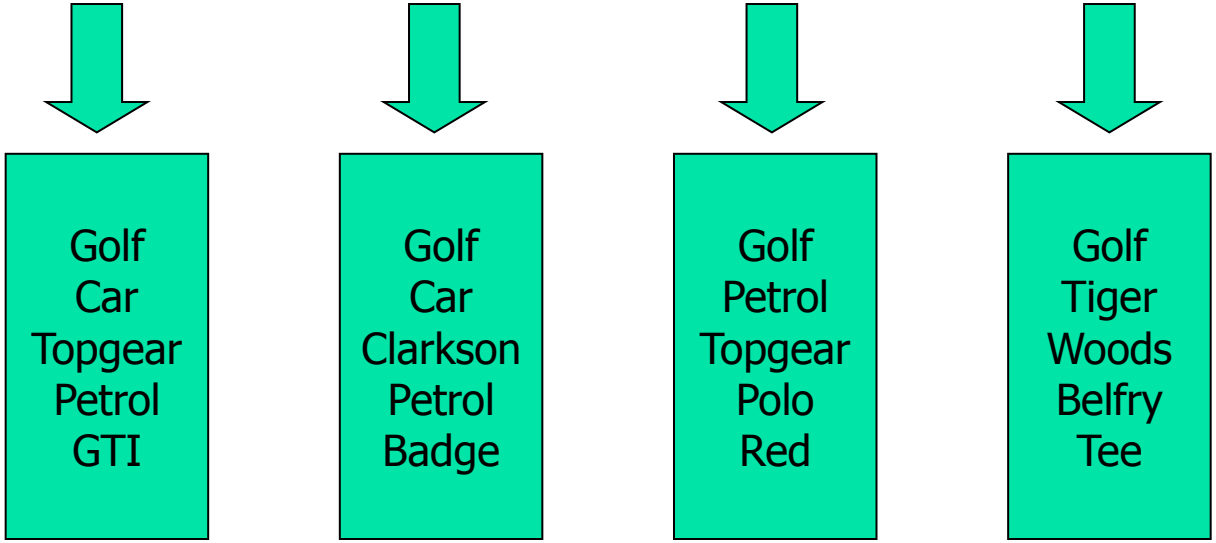


Base di documenti (20)

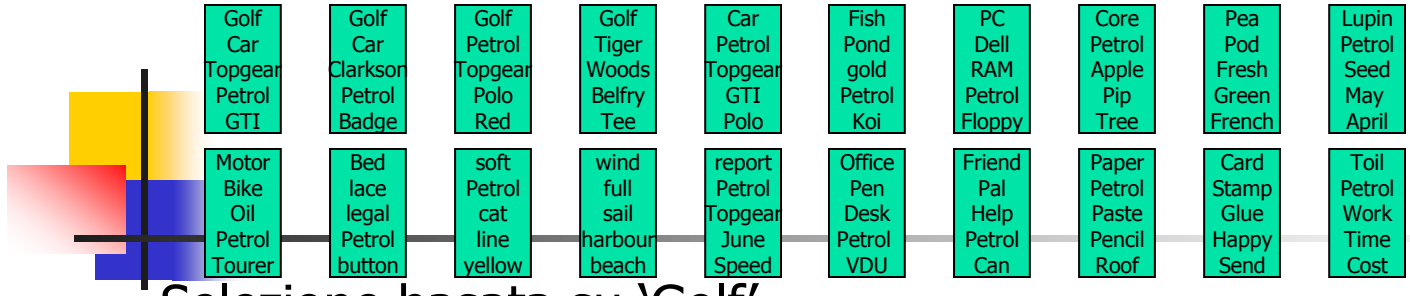


con il modello keyword vengono estratti 4 documenti

Selezione dei documenti basata sul termine 'Golf'



Tutti i 20 documenti



Selezione basata su 'Golf'

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Golf Petrol Topgear Polo Red

Golf Tiger Woods Belfry Tee

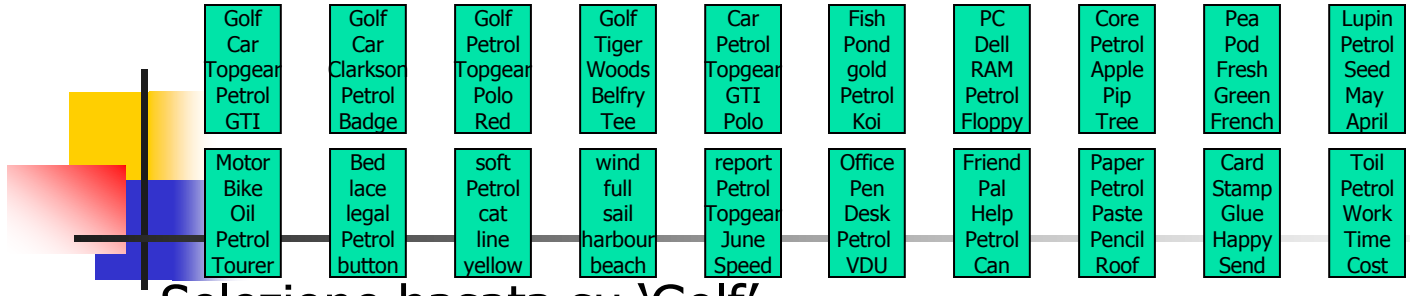
rank dei doc selezionati

wf.idf

Car
2 *(20/3) = 13
Topgear
2 *(20/3) = 13
Petrol
3 *(20/16) = 4

vediamo quali sono le parole più rilevanti associate a Golf di questi 4 documenti. Esse sono: Car, Topgear and Petrol

Tutti i 20 documenti



Selezione basata su 'Golf'

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Golf Petrol Topgear Polo Red

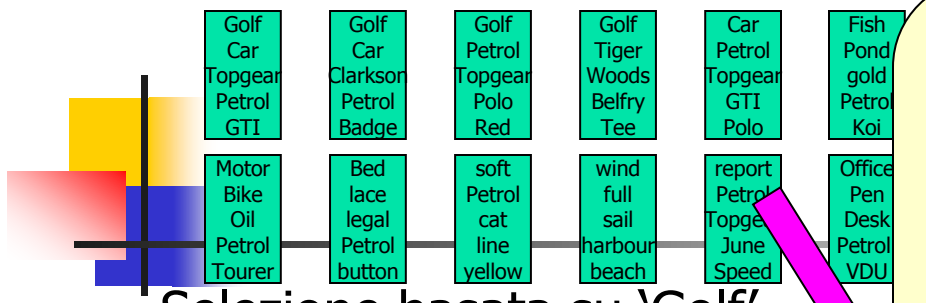
Golf Tiger Woods Belfry Tee

rank dei doc selezionati

Car
2 *(20/3) = 13
Topgear
2 *(20/3) = 13
Petrol
3 *(20/16) = 4

poiché le parole sono pesate anche rispetto al loro idf, risulta che :
Car e Topgear sono associate a Golf più di Petrol

Tutti i 20 documenti



Selezione basata su 'Golf'

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Golf Petrol Topgear Polo Red

selezione basata sul dominio semantico

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Golf Petrol Topgear Polo Red

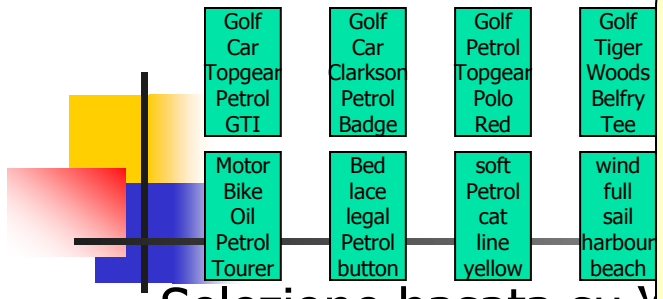
Golf Tiger Woods Belfry Tee

Car Wheel Topgear GTI Polo

Ora cerchiamo ancora nella base di documenti, usando questo insieme di parole che rappresentano il "**dominio semantico**" di Golf .
La lista ora include un nuovo documento, non catturato sulla base della semplice ricerca per keywords.

Car
2 *(20/3) = 13
Topgear
2 *(20/3) = 13
Petrol
3 *(20/16) = 4

Tutti i 20 documenti



Usando un ranking
basato sulla co-occorrenza dei termini
possiamo assegnare
un miglior ranking ai documenti.
Notate che: il documento
più rilevante non contiene la parola Golf, e
che uno dei documenti che la conteneva
scompare (era infatti un senso "spurio"di Golf).

Selezione basata su '

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Petrol Topgear Polo Red

Car
2 *(20/3) = 13
Topgear
2 *(20/3) = 13
Petrol
3 *(20/16) = 4

selezione basata sul dominio semantico

Golf Car Topgear Petrol GTI

Golf Car Clarkson Petrol Badge

Golf Petrol Topgear Polo Red

Golf Tiger Woods Belfry Tee

Car Wheel Topgear GTI Polo

Rank

30

17

17

0

26

Relevance Feedback

- Il **Relevance Feedback** e **Query Expansion** sono tecniche per migliorare il recall di una query.
- Nel Relevance Feedback l'idea è che dopo la presentazione di un set iniziale di documenti, si chiede all'utente di selezionare i più rilevanti, usando questo feedback per riformulare la query, nuovamente quindi si presentano nuovi risultati all'utente, eventualmente, iterando il processo.
- Nella Query expansion, si aggiungono termini oltre quelli iniziali, con l'obiettivo di migliorare la qualità della ricerca

Relevance Feedback

- Come tener conto del feedback?
  - –Query Expansion: Aggiungi alla query **nuovi termini** estratti dai documenti prescelti dall'utente
  - –Term Reweighting: Aumenta il peso dei termini che compaiono nei documenti rilevanti e diminuisci il peso di quelli che non vi compaiono.
- Diversi algoritmi per effettuare la riformulazione della query.

Relevance Feedback

Nel modello vettoriale, si modifica il vettore della query **Aggiungendo i vettori dei documenti rilevanti** al vettore della query e **sottraendo i vettori dei documenti irrelevanti al** vettore della query. La totalità dei documenti rilevanti non è tuttavia nota, per cui si operano delle approssimazioni, ad esempio conoscendo solo, fra quelli proposti all'utente, la frazione dei rilevanti (Dr) e irrelevanti (Dn) rispetto alla query iniziale q, si perviene alla **Formula di Rocchio:**

$$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

α: Un peso (regolabile) per la query iniziale.
β: peso dei documenti rilevanti.
γ: peso dei documenti irrilevanti.

Relevance Feedback

Il feedback esplicito non è molto usato:

- Gli utenti sono a volte riluttanti.
- E' più difficile capire perché un documento sia stato selezionato (l'utente può rendersi conto di aver mal formulato la query e le sue selezioni appaiono inconsistenti con i primi risultati proposti).

Per questo motivo si introduce il **Pseudo Feedback**:

- Non chiedere esplicito aiuto all'utente.
- Assumi che i primi m top-ranked siano i più interessanti.
- Espandi la query includendo termini correlati con i termini della query, usando gli m top-ranked.
- Il metodo si è dimostrato efficace

Query Expansion

La Query expansion incrementa i termini utilizzati nella query, e a tale scopo può attingere da un glossario (**thesaurus**), che fornisce informazioni di sinonimia e correlazione fra termini (ad esempio WordNet) oppure utilizzare le **co-occorrenze**.

Ma Iponimi e sinonimi migliorano la Ricerca? In generale:

–NO se i termini della ricerca sono pochi e poco specifici (ambiguità genera rumore)

–SI se i termini non sono ambigui (domini tecnici)

–NI se si applicano algoritmi di word sense disambiguation:

SI per query lunghe (molto "contesto" migliora WSD)

NO per query corte e generiche (poca precisione nella disambiguazione)

Query Expansion

- Usando un thesaurus, Per ogni termine t, in una query, si espande la query con sinonimi e termini correlati nel thesaurus.
- In genere i pesi dei termini aggiunti sono più bassi.
- Questo metodo aumenta la recall ma diminuisce la precisione, per via dell'ambiguità semantica (aggiungere sinonimi con AND in Google quindi in genere peggiora, un po' meglio se si espande con OR)

Query Expansion

- Nel caso delle co-occorrenze, si determina anzitutto la similarità fra termini usando delle statistiche pre-calcolate sull'intera collezione di documenti (global analysis). Si calcolano matrici associative (matrici di co-occorrenze) che quantificano la correlazione fra termini. Si espande infine la query con i termini più simili.
- Poiché i termini sono in ogni caso altamente correlati, l'espansione potrebbe non aggiungere molti nuovi documenti!
- Se L'analisi dei termini correlati non è basata sull'intera collezione, ma solo sui documenti "localmente" recuperati sulla base della query iniziale, si parla di local analysis. Questo riduce il problema della ambiguità semantica, perché i documenti, essendo recuperati solo localmente, molto probabilmente contengono ogni termine nel senso corretto per l'utente
- L'analisi globale richiede di fare dei calcoli una volta per tutte, l'analisi locale va fatta in tempo reale sulla base di ogni query ma fornisce risultati migliori.

Query Expansion

Google query expansion opera con:

- Word stemming: translator -> translator, translation
- Acronimi: NATO -> North Atlantic Treaty Organization (pericoloso… Northern Arts Tactical Offensive)
- Errori di digitazione: wigets ->widgets
- Sinonimi: solo se appare evidente che la parola è usata in modo improprio (information lost ->loss)
- Traduzione (organizzazione mondiale sanità -> world health organization)
- Related Search (migliorata dal 2009 dopo l'accordo con Orion)