



Opportunità, innovazione e rischi nell'era dell'Intelligenza Artificiale

Information Retrieval
Intervento

Ing. Claudio Santoro

Ordine degli Ingegneri della Provincia di Roma

Roma, 22 Maggio 2025



Ing. Claudio Santoro

Prima laurea (VO): Ingegneria Informatica
Seconda laurea (Magistrale): SMFN Informatica

Certificazioni: Prince2®- ITIL®
Master (secondo livello): BIM&GIS

Dal 10.2012 ad oggi: PA
Dal 11.2003: Privato

Oltre 50 progetti (metà internazionali)



INFORMATION RETRIEVAL

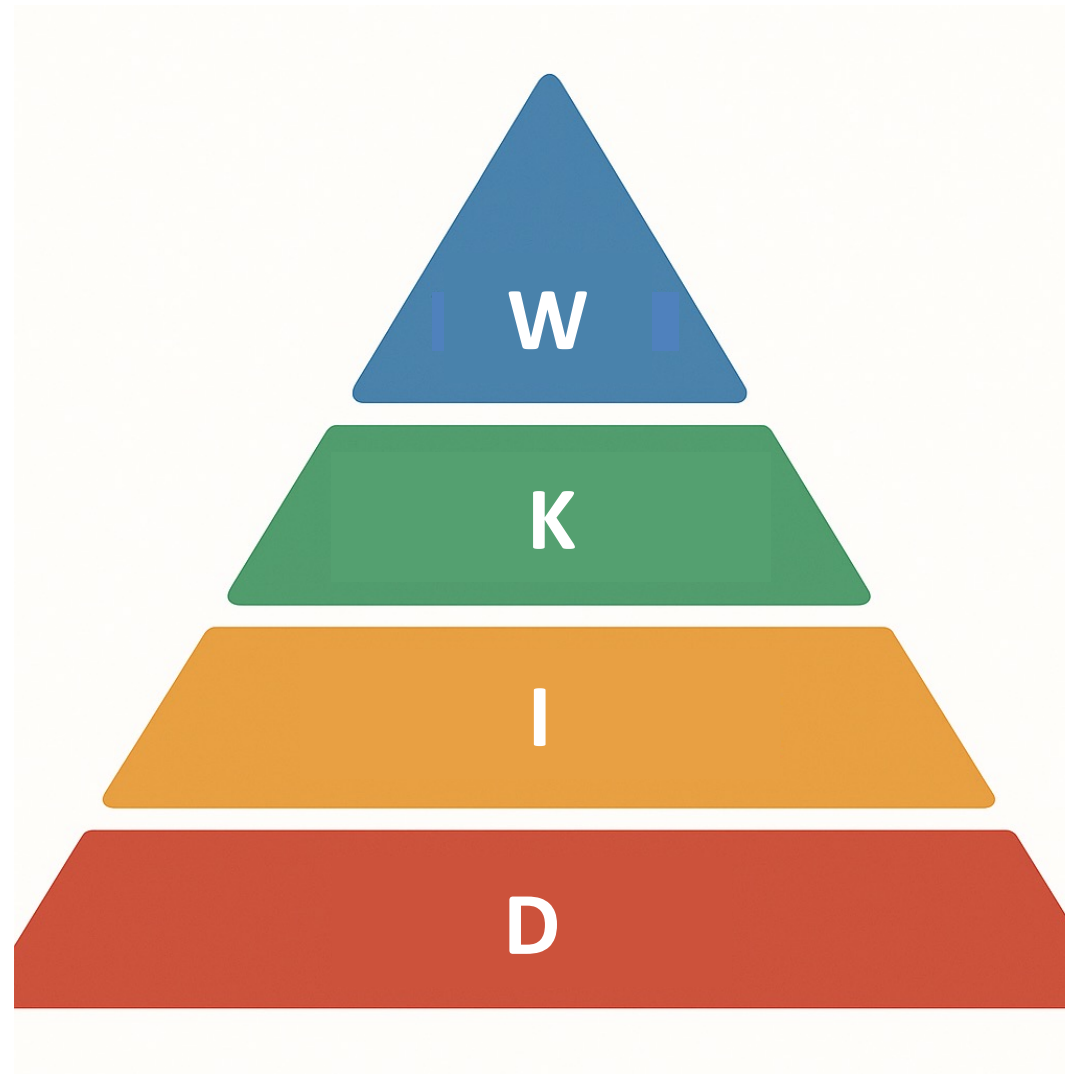


Information: Definizione

L'Information Retrieval (IR) si occupa della rappresentazione, memorizzazione e organizzazione dell'**informazione testuale**, al fine di rendere agevole all'utente il **soddisfacimento** dei propri **bisogni informativi**.



Information: Piramide Dijkstra





Information: Piramide DIKW



glos	39026121493
eliasson	39068536972
saeed	39027436851



Information: Piramide DIKW

Enigma:

- Brevetto 1918
- Prima guerra mondiale: comunicazioni navali Tedesche decriptate dai Britannici
- utilizzata dai Tedeschi durante la Seconda guerra mondiale
- 1932 matematici Polacchi decifrarono «garantendo **informazione** alle truppe alleate»



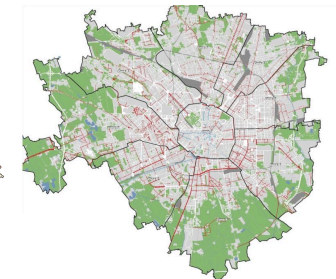
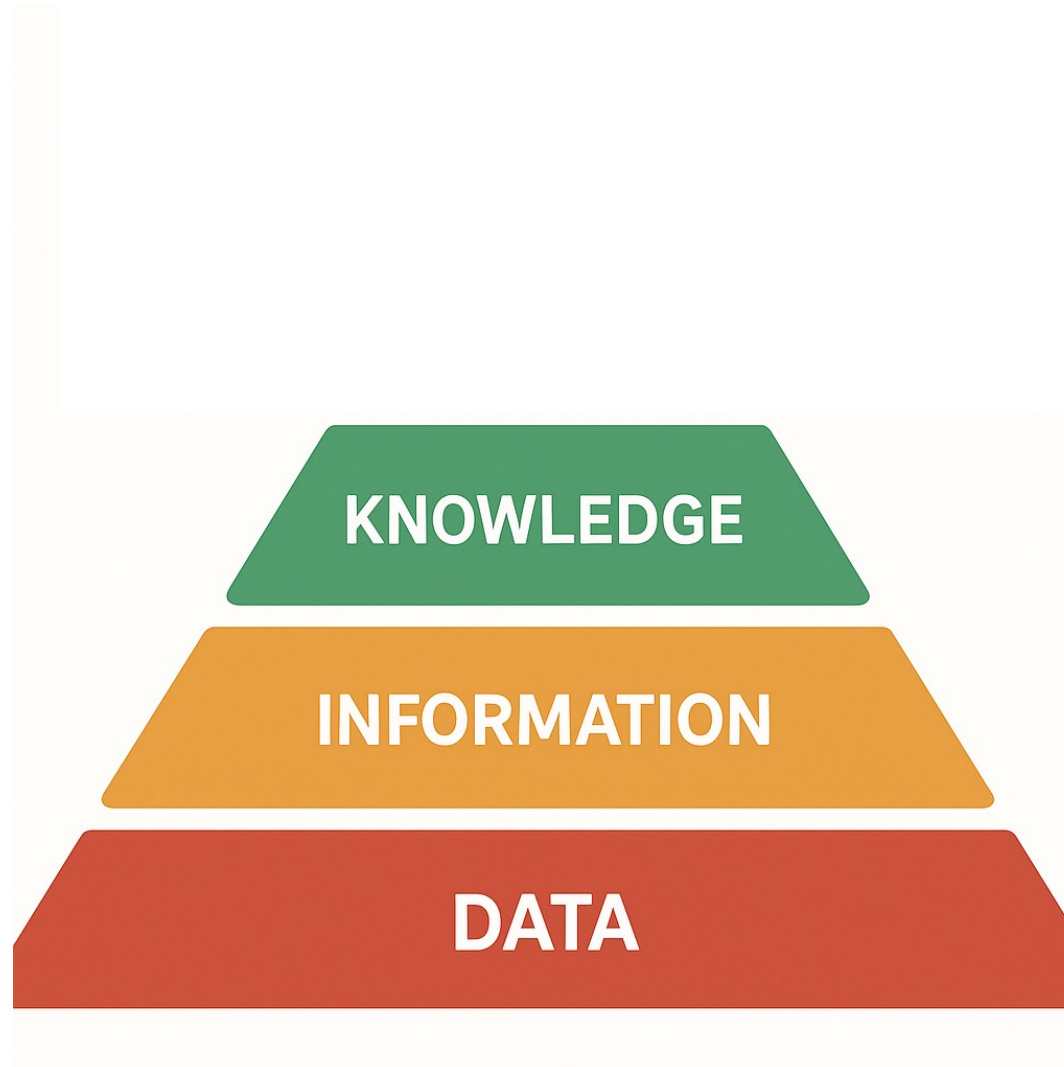
INFORMATION

DATA

NOME	TELEFONO
glos	39026121493
eliasson	39068536972
saeed	39027436851



Information: Piramide DIKW



NOME	TELEFONO
glos	39026121493
eliasson	39068536972
saeed	39027436851



Perché «Piramide»?



Retrieval: Interrogazione layer Data





Retrieval: Interrogazione layer Data

Query - postgres on postgres@localhost:5432

File Edit Query Favourites Macros View Help

Previous queries Delete Delete All

```
SELECT product_id, description, cost, price FROM products
WHERE price >= 10.00 AND price <=20.00
```

Output pane

Data Output Explain Messages History

	product_id character varying(5)	description text	cost numeric(5,2)	price numeric(5,2)
1	A001	Greek Salad	9.54	16.45
2	A003	Mediterranean Salad	9.54	16.45
3	A005	Turkish Salad	3.54	16.45
4	Q002	Pecan Nuts	8.54	15.50

OK. DOS Ln 24, Col 1, Ch 693 4 rows. 13 msec



Retrieval: Interrogazione layer Information

The screenshot shows a Google search interface with four search queries entered in the search bar: "dolore al ginocchio, ginocchio gonfio", "cosa fare in caso di ginocchio gonfio", "fisioterapia in caso di ginocchio gonfio", and "tecarterapia per ginocchio gonfio". The search results for "tecarterapia per ginocchio gonfio" are displayed, showing a list of results under the "Tutti" tab. The first result is from Fisioglobe, dated 3 giu 2022, titled "Tecarterapia al ginocchio, quando farla", and includes a thumbnail image of a person receiving treatment. Below this result is a section titled "Le persone hanno chiesto anche" with four related questions: "La tecarterapia è efficace per il liquido al ginocchio?", "Come sfiammare un ginocchio gonfio?", "La Tecar toglie infiammazione?", and "Quale fisioterapia per ginocchio gonfio?". The second result is from fisioterapiabrotini.it, dated 1 feb 2023, titled "La Tecarterapia nelle patologie del ginocchio", and includes a thumbnail image of a person receiving treatment.

Google

dolore al ginocchio, ginocchio gonfio

Google

cosa fare in caso di ginocchio gonfio

Google

fisioterapia in caso di ginocchio gonfio

Google

tecarterapia per ginocchio gonfio

Tutti Immagini Video Video brevi Libri Web Notizie Altro

Strumenti

Fisioglobe
https://www.fisioglobe.it > Il Blog della Fisioterapia

Tecarterapia al ginocchio, quando farla

3 giu 2022 — La Tecarterapia è utilizzata per ridurre gli ematomi successivi ai traumi distorsivi del ginocchio. Sia per le lesioni parziali del legamento ...

Le persone hanno chiesto anche

La tecarterapia è efficace per il liquido al ginocchio?

Come sfiammare un ginocchio gonfio?

La Tecar toglie infiammazione?

Quale fisioterapia per ginocchio gonfio?

Feedback

fisioterapiabrotini.it
https://www.fisioterapiabrotini.it > post > la-tecarterapia...

La Tecarterapia nelle patologie del ginocchio

1 feb 2023 — La tecarterapia risulta essere efficace per migliorare il drenaggio dell'articolazione in seguito a presenza di edema e/o versamento, post ...

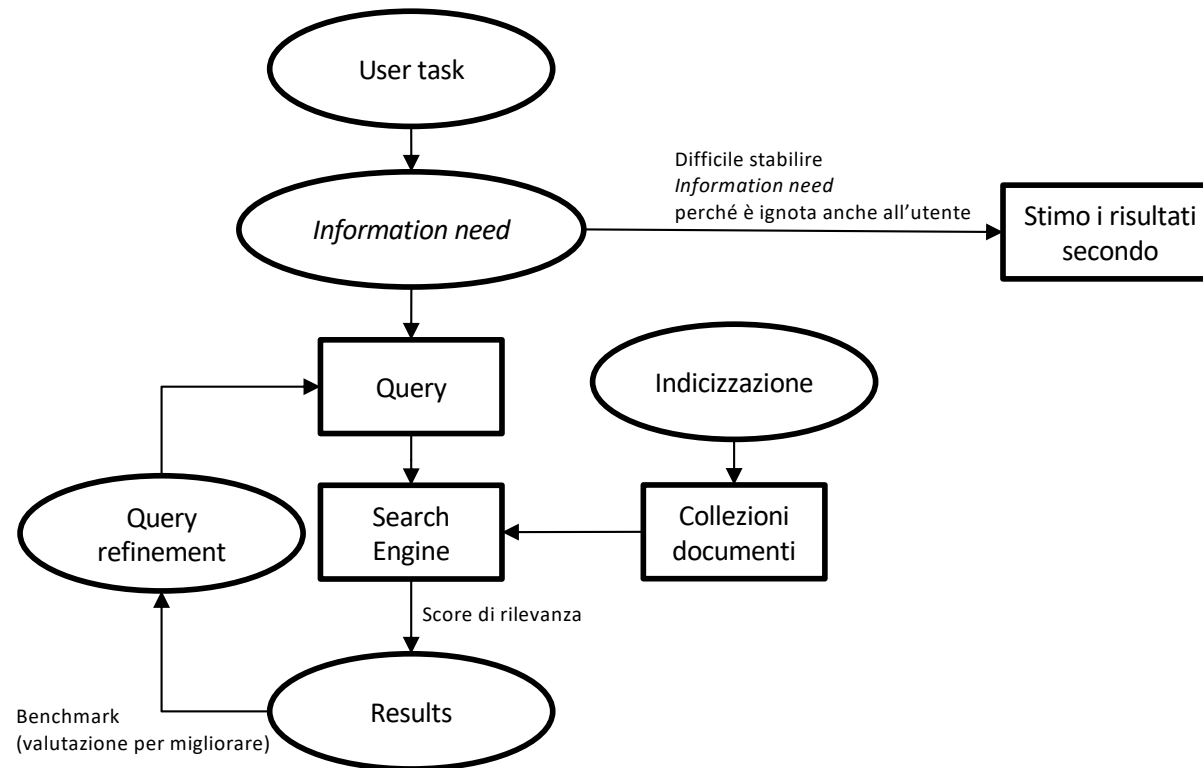


Retrieval: Interrogazione

- Conoscenza del dominio
 - Modalità di formalizzazione della richiesta
 - Natura del risultato



Information Retrieval





Information Retrieval: Boolean retrieval model

Boolean Retrieval Model: Matrice termini per documenti

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Righe: Termini (TOKEN)

Colonne: Documenti

(1 se presente)

Query (Boolean Retrieval Model)

Esempio: Brutus AND Caesar AND NOT Calpurnia

Brutus AND ▪ 110100 AND
Caesar AND ▪ 110111 AND
NOT Calpurnia ▪ 101111 =
 ▪ 100100

Pro	Contro
Veloce	- Indice molto grande - Matrice composta in prevalenza da 0



Soluzione 1: uso Notazione sparsa per comprimere matrice: **Inverted Index**

OSS: **Notazione Densa**: riporto 0 e 1

Notazione Sparsa: riporto solo valore non nulli



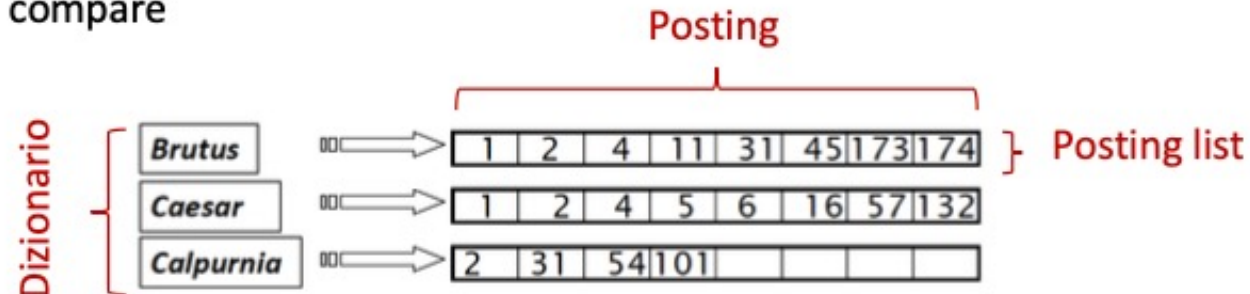
Information Retrieval: Inverted index model

Inverted Index

- **Struttura**

1/2: Ad ogni documento associo un ID

2/2: per ogni termine ho una lista ordinata di documenti in cui quel termine compare



Nota:

- Posting list è **ordinata** e di **dimensione variabile**
- Lista è **sequenza di blocchi contigui** allocati staticamente

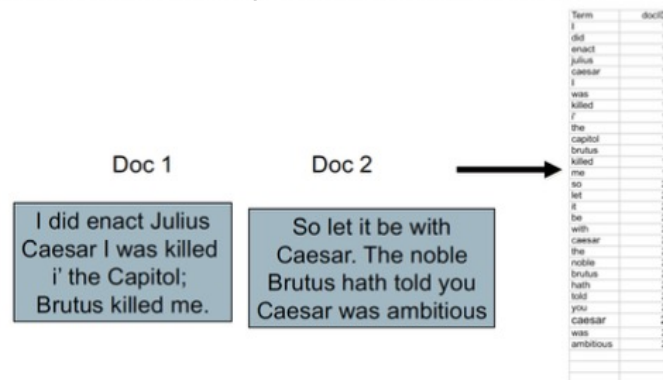


Information Retrieval: Inverted index model

• Costruzione Indice

1/3: **Token sequence**: inserisco i token accodandoli

Nota: struttura hashmap -> costo di inserimento *cost*



2/3: **Sort**: ordino alfabeticamente i token

Nota: Costo quick sort $n \log n$

Term	docID	Term	docID
I	1	ambitious	2
did	1	be	2
enact	1	Brutus	1
Julius	1	Brutus	2
Caesar	1	Capitol	1
I	1	Caesar	1
was	1	Caesar	2
killed	1	Caesar	2
i'	1	did	1
the	1	enact	1
Capitol	1	hath	1
Brutus	1	I	1
killed	1	I	1
me	1	i'	1
so	2	it	2
let	2	Julius	1
it	2	killed	1
be	2	killed	1
with	2	let	2
Caesar	2	me	1
the	2	noble	2
noble	2	so	2
Brutus	2	the	1
hath	2	the	2
told	2	told	2
you	2	you	2
Caesar	2	was	1
was	2	was	2
ambitious	2	with	2

OSS: Ordino (sort) per fare poi il

Merge necessario a fare query tra più termini (altrimenti per un Term, basta sfilare la posting list)

3/3: **Reduce**: riduco e inserisco la Document frequency

Nota: Costo di accesso a termine -> *cost* (hashmap)

Costo scansione posting list -> *cost*

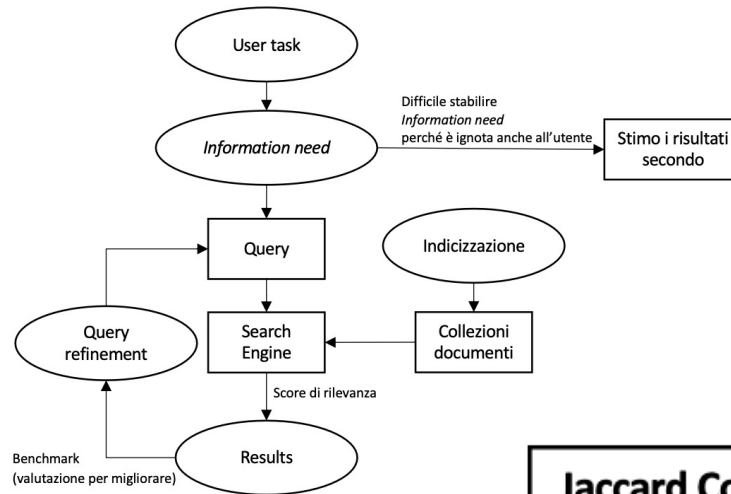
Term	doc	freq.	postings lists
ambitious	1	1	2
be	1	1	2
Brutus	2	2	1 → 2
Capitol	1	1	1
Caesar	2	2	1 → 2
did	1	1	1
enact	1	1	1
hath	1	1	2
i	1	1	1
i'	1	1	1
it	1	1	2
Julius	1	1	1
killed	1	1	1
let	1	1	2
me	1	1	1
noble	1	1	2
so	1	1	2
the	2	2	1 → 2
told	2	1	2
you	1	1	2
was	2	2	1 → 2
with	1	1	2

OSS: **Document frequency**: su quanti documenti appare il termine
Term frequency: quante volte appare il termine nel documento

Oss: Phrase queries



Information Retrieval: Jaccard



Jaccard Coefficient

Idea:

- Vedo la **query** come un documento composto da termini
- Misuro la sovrapposizione dei termini con i termini del **documento**



$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

• Infatti:

- Score compreso tra 0 e 1
- $Jaccard(A, A) = 1$
- $Jaccard(A, B) = 0$ se $A \cap B = \emptyset$



Information Retrieval: Term frequency

Count matrix

- ogni documento è vettore **count** di termini
 - Righe: termini
 - Colonne: documenti
 - Incremento di «1» ogni volta che trovo il termine nel documento

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	157	73	0	0	0	1	
BRUTUS	4	157	0	2	0	0	
CAESAR	232	227	0	2	1	0	
CALPURNIA	0	10	0	0	0	0	
CLEOPATRA	57	0	0	0	0	0	
MERCY	2	0	3	8	5	8	
WORSER	2	0	1	1	1	5	
...							

Nota:

- Modello **bag of words**: non mi interessa l'ordine in cui i termini appaiono ma solo la loro presenza

Term Frequency

- Definisco Term Frequency $tf_{t,d}$ del termine t nel documento d , come il numero di volte in cui t occorre in d .



Information Retrieval: Inverse document frequency

Inverse Document Frequency

Quindi, definisco **Inverse Document Frequency**:

$$\text{idf}_t = \log \frac{N}{\text{df}_t}$$

#Documenti dove
termine appare meno
volte nella Collezione
(più raro -> più
rappresentativo)

Dove:

- N : numero documenti nella collezione
- df_t : numero di documenti in cui il termine compare

Quindi

- più idf_t è alto, più il termine è «raro» rispetto la collezione;
- idf_t fa pesare un termine più dell'altro -> ha senso con query con più termini



Information Retrieval: Weight matrix

tf-idf weight

$$w_{t,d} := (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

Peso la frequenza del Termine,
considerando la sua rarità nella
Collezione

Nota:

- Aumenta:
 - Con l'aumentare del Term Frequency: $\text{tf}_{t,d}$ (ovvero il numero di volte in cui un Termine appare nel Documento)
 - Con l'aumentare dell' Inverse Document Frequency: idf_t (ovvero l'inverso del numero dei documenti che contengono il termine t rispetto l'intera collezione – documenti dove è più raro)

Weight Matrix Usage

Premessa:

- Uso quindi tf-idf
- Possiamo pre-calcolarla perché non necessita riferimento alla query ma solo l'analisi della Collezione

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	5.25	3.18	0.0	0.0	0.0	0.35	
BRUTUS	1.21	6.10	0.0	1.0	0.0	0.0	
CAESAR	8.59	2.54	0.0	1.51	0.25	0.0	
CALPURNIA	0.0	1.54	0.0	0.0	0.0	0.0	
CLEOPATRA	2.85	0.0	0.0	0.0	0.0	0.0	
MERCY	1.51	0.0	1.90	0.12	5.25	0.88	
WORSER	1.37	0.0	0.11	4.15	0.25	1.95	
...							

Come rispondiamo a una query?

Prima soluzione: Sommiamo i valori dei vari termini della query, presente nei Documenti e il maggiore avrà score più alto

Problema:

- Non teniamo in considerazione la lunghezza dei documenti
- Così facendo un documento più lungo avrà più probabilità di avere score maggiore



Information Retrieval: Vector space model

Vector Space Model

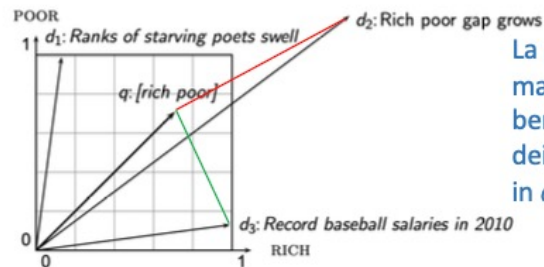
Modello i documenti in uno spazio vettoriale e la similarità è data dal cos dell'angolo sotteso (come calcolo le componenti dei vettori è un'altra questione)

Idea:

- V : ogni documento è un vettore di termini pesati secondo tf-idf
- Pertanto lo spazio $|V|$ sono le dimensione dei termini e i documenti sono punti o vettori in tale spazio
- Pertanto lo score di un documento sarà pari alla «somiglianza» di questo con il documento query

Soluzione 1: Distanza Euclidea

- Considero la distanza euclidea tra vettori
- Non ci piace perché è alta con vettori di differente lunghezza



La distanza rossa è maggiore della verde benché la distribuzione dei termini sia migliore in d_2

The Euclidean distance of $\vec{q}$ and $\vec{d}_2$ is large although the distribution of terms in the query q and the distribution of terms in the document d_2 are very similar.



Information Retrieval: Probabilistico

Ora usiamo un approccio probabilistico:

- Variabile booleana esterna che ci dice se documento è rilevante (o meno) per una data query

Pertanto il **Rilevanza** $R_{d,q}$:

$$R_{d,q} := \begin{cases} 1 & , \text{ if document } d \text{ is relevant for query } q \\ 0 & , \text{ else} \end{cases}$$

Ovvero $P(R = 1|d, q)$ Probabilità che la Rilevanza sia uguale ad 1, sapendo d e q

Pertanto si assume che **la rilevanza di un documento sia indipendente dalla rilevanza degli altri documenti.**



Probabilistic Ranking Principle

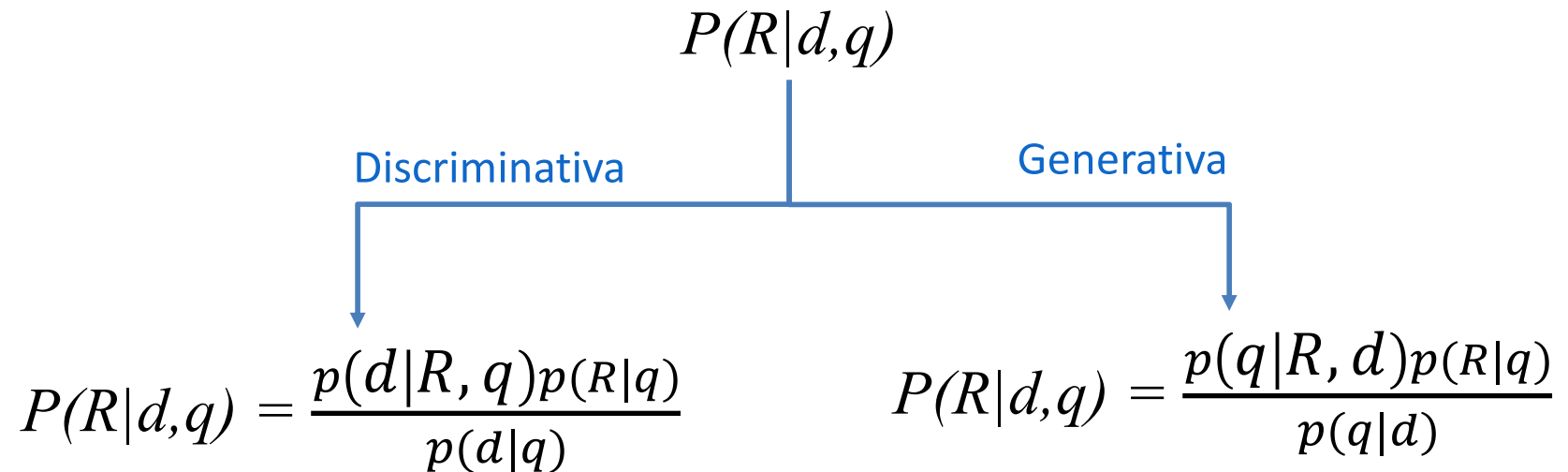
Se ordino i documenti in base alla probabilità di essere rilevanti rispetto la query -> sistema si comporta nel modo migliore

—



Information Retrieval: Probabilistico

$$P(A|B, C) = \frac{P(B|A, C)P(A|C)}{P(B|C)}$$





Information Retrieval: Probabilistico

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

Modelli di Linguaggio

1. Da d passo al modello M_d
2. Bayes:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

Probabilità a priori
Non mi interessa perché costante in tutti i documenti

$$P(q|M_d) = P(t_1, t_2, \dots, t_{|q|}|M_d)$$

3. **1° HP Naive Bayes** (indipendenza dei termini della query):

$$\begin{aligned} P(q|M_d) &= P(t_1|M_d)P(t_2|M_d) \dots P(t_{|q|}|M_d) = \\ &= \prod_{1 \leq k \leq |q|} P(t_k|M_d) \end{aligned}$$

4. Per stimare $P(t_k|M_d)$ uso **Massima Verosimiglianza**: considero term frequency (ovvero tutti i termini che sono nella query, rispetto a tutti i termini del Modello di Linguaggio del documento d) sul numero dei termini del documento

$$\hat{P}(t|M_d) = \frac{tf_{t,d}}{|d|}$$

#Termine appare nel Documento / Totale termini documento

