Università degli Studi di Cagliari
Corso di Laurea DSBAI

Web Analytics e Analisi Testuale

http://agilegroup.eu

A.A. 2019/2020

Ing. Marco Ortu
Via Porcell 4, primo piano
mail: marco.ortu@unica.it

Information Retrieval e Fondamenti di Web e Text Search

Motori di ricerca Web

I web search engine condividono parte delle problematiche dei generici sistemi di IR, ma hanno specificità notevoli e utilizzano pertanto anche altre tecniche.

Principali differenze: volatilità delle risorse (non possono essere controllate); assenza di un vocabolario definito;

- ➔ numero enorme di risorse; risorse statiche e generate
- ➔ dinamicamente; formati diversi (pdf, html, etc.) ;

difficoltà dell'utente nell'individuare il proprio bisogno informativo; utilizzo di tecniche di ranking non basate esclusivamente sul contenuto (e.g. pagerank); uso di spider e crawler per effettuare la ricerca delle risorse.

Meta informazioni nel Web

Normalmente l'indicizzazione e la ricerca vengono svolte sul contenuto del documento

L'utente può accedere e leggere le stesse parole usate per la creazione degli indici e per la ricerca

➔ Il documento descrive sé stesso, in base alle parole che lo formano

Le pagine HTML possono contenere dati, non visibili all'utente, che descrivono esplicitamente il contenuto della pagina Web

Ci si riferisce a queste informazioni aggiuntive con il termine di meta-informazioni

➔ Le meta-informazioni possono riguardare: descrizione con parole chiave, lingua utilizzata, autore, data di creazione e modifica

I motori di ricerca nel Web possono utilizzare le parole chiave contenute tra le meta-informazioni, per migliorare le prestazioni

Altri approcci alla ricerca nel Web

L'utilizzo dei SE non è intuitivo per gli utenti inesperti, perciò sono disponibili altri modi per reperire informazioni nel Web

Directory

➔ Gestisce solo pagine che sono state scelte attraverso un processo di selezione/catalogazione editoriale o sottoposte dagli stessi utenti http://www.yahoo.com

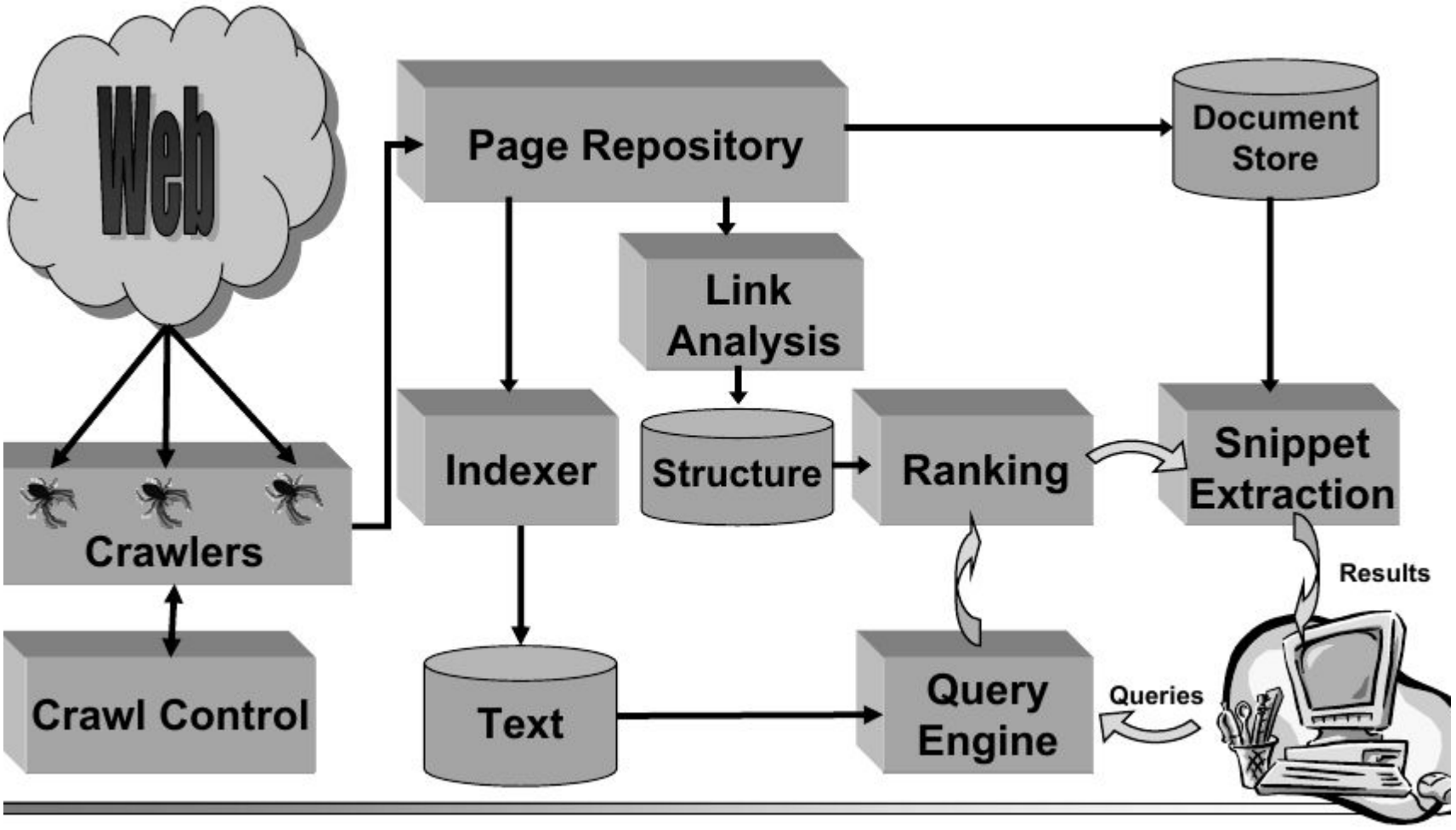
Meta Search Engine

➔ Uno strumento che interroga contemporaneamente diversi SE e/o directory e riassume i risultati all'utente http://www.metacrawler.com

Portali

➔ Non sono dei reali sistemi per il reperimento dell'informazione, ma si presentano come punti di partenza per la navigazione http://www.virgilio.it

Architettura di un Search Engine



Crawler e spider

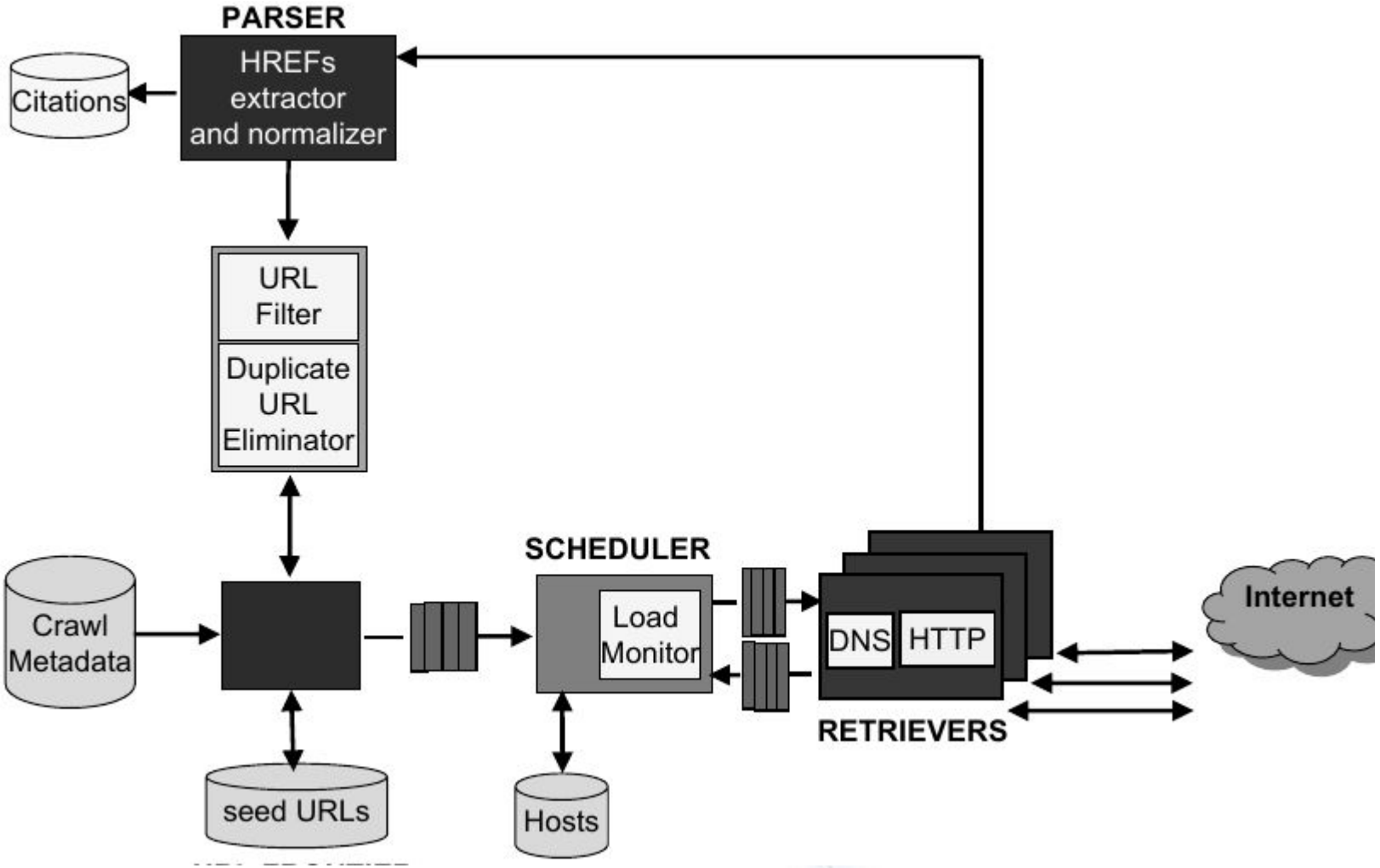
Attività: navigare il web prelevando le risorse (web page, documenti vari) che verranno indicizzate.

Problematiche: ricercare il più possibile; ricercare risorse rilevanti; non seguire più volte il medesimo cammino di ricerca; bilanciare il carico e i percorsi tra i vari crawler che lavorano in parallelo; evitare i mirror di medesime risorse.

Ulteriori problematiche: non sovraccaricare l server con richieste (rispettare netiquette); gestire il refresh di risorse già indicizzate (le pagine possono cambiare nel tempo)

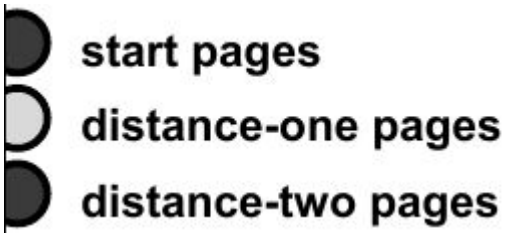
Vari nomi: crawler, spider, bot (abbrv. robot)

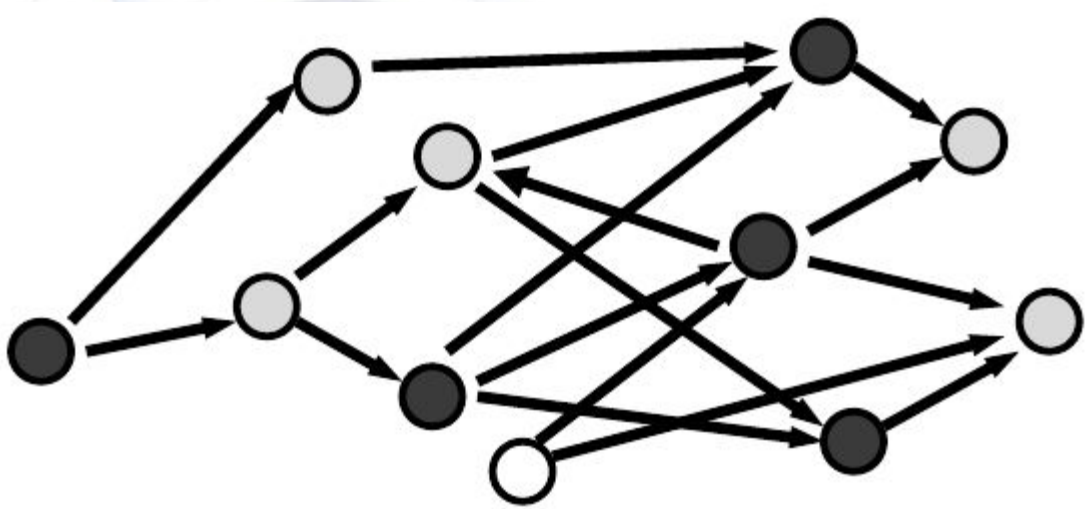
Architettura di un Crawler



Ricerca Breadth-First

Iniziare con un insieme di URL note (semi) Ricercare in “cerchi concentrici” intorno a queste URL





Usato da molti crawler

Aiuta a bilanciare il carico

Alternativo al **Depth-first**

Regolamentare l’accesso ad un server

Il file robots.txt può essere usato per impedire/regolamentare l’accesso a agli spider

Costituito da coppie: user-agent: textstring (il crawler da regolamentare) disallow:textstring (le risorse il cui accesso va regolamentato)

Esempio:

er-agent: *

Esempio:
User-agent: *
Disallow: /cgi-bin/ (impedisce a tutti I crawler -* è una wild card l’accesso alla directory specificata)
User-agent: googlebot
Disallow: * (impedisce l’accesso al crawler di google)

Indicizzazione web

La prima generazione di web search engine utilizzava esclusivamente le tecniche classiche di IR.
Attualmente a queste si aggiungono dati web-specifici non dipendenti dal contenuto delle pagine (approccio Google-like).

In particolare:

- ➔ Link (or connectivity) analysis (I link rappresentano “voti” sulla rilevanza di una pagina)
- ➔ Click-through data (su quali risultati di una ricerca gli utenti cliccano)
- ➔ Anchor-text (come gli utenti si riferiscono alla risorsa)

Ordinamento Query-independent

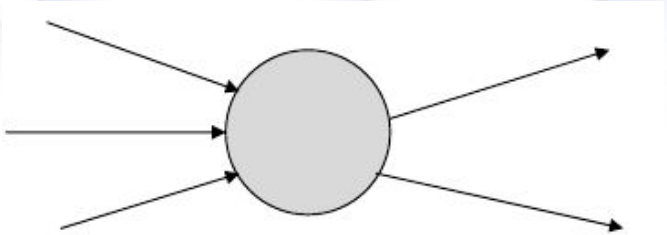
Tecnica base: usare il numero di link come misura di popolarità (molte citazioni = importante)

➔ Undirected popularity:

- Per ogni pagina: **score** = il numero di in-links più il numero di out-links (3+2=5)

➔ Directed popularity:

- Per ogni pagina: **score** = il numero di in-links (3)



Nuova logica del Query processing

1. Recuperare le risorse che verificano la query usando IR classico

2. Pesa e ordina le risorse usando la link popularity

3. Limite: approccio semplice, ma prono allo spam

Criteri di ordinamento dei documenti

La **pesatura** dei termini consente di ordinare i documenti

- ➔ Termine frequente nel documento = peso maggiore
- ➔ Termine frequente nella collezione = peso minore

Ci sono altre strategie che dipendono dalla **struttura** del testo

- ➔ Termini nel tag <title> o nell'URL = peso maggiore
- ➔ Termini nei tag H1-H6 = peso maggiore
- ➔ Termini all'inizio del testo = peso maggiore
- ➔ Termini vicini tra loro = rinforzo reciproco del peso

E' inoltre importante **l'autorevolezza** della pagina

- ➔ Misurata in base al numero di pagine che hanno un link
- ➔ Ogni pagina ha una sua autorevolezza che ridistribuisce alle pagine verso cui ha un link
- ➔ Bisogna essere puntati da pagine a loro volta autorevoli

PageRank

Cerca di catturare la nozione di **“rilevanza”** di una pagina (indipendentemente dal suo contenuto)

Usa i **Backlinks** per il ranking

Evita lo **spamming**: Distribuisce “il voto” delle pagine tra quelle a cui esse sono collegate

Pagine **“Importanti**” che puntano ad una pagina innalzeranno il rank
di quest’ultima più di altre

Nota: Google usa il pagerank, ma anche numerose altre euristiche
non pubbliche.

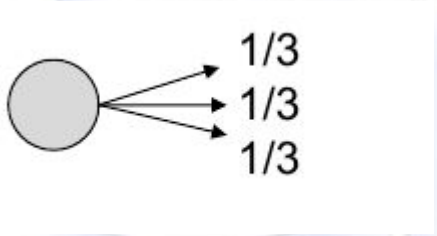
PageRank: scoring

Si considera un browser che effettua una **“random walk”** tra le pagine:

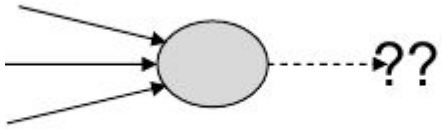
- ➔ Inizia con una pagina random
- ➔ Ad ogni passo, segue uno dei link in maniera equiprobabile

“All’equilibrio” ogni pagina ha uno score a lungo termine di visite: questo è il suo page score

- ➔ 1/3
- ➔ 1/3
- ➔ 1/3



• Problema: I vicoli ciechi (blind alleys)



PageRank: Teleporting

- ➔ Ad ogni passo, con probabilità 10%, scegli una web page in maniera random
- ➔ con probabilità 90%, segui uno dei link in maniera random
- ➔ Ciò evita di rimanere bloccati nei vicoli ciechi

PageRank: Base

$$PR[A] = \frac{1-d}{N} + d\left(\sum_{k=1}^{n} \frac{PR[P_k]}{C[P_k]}\right)$$

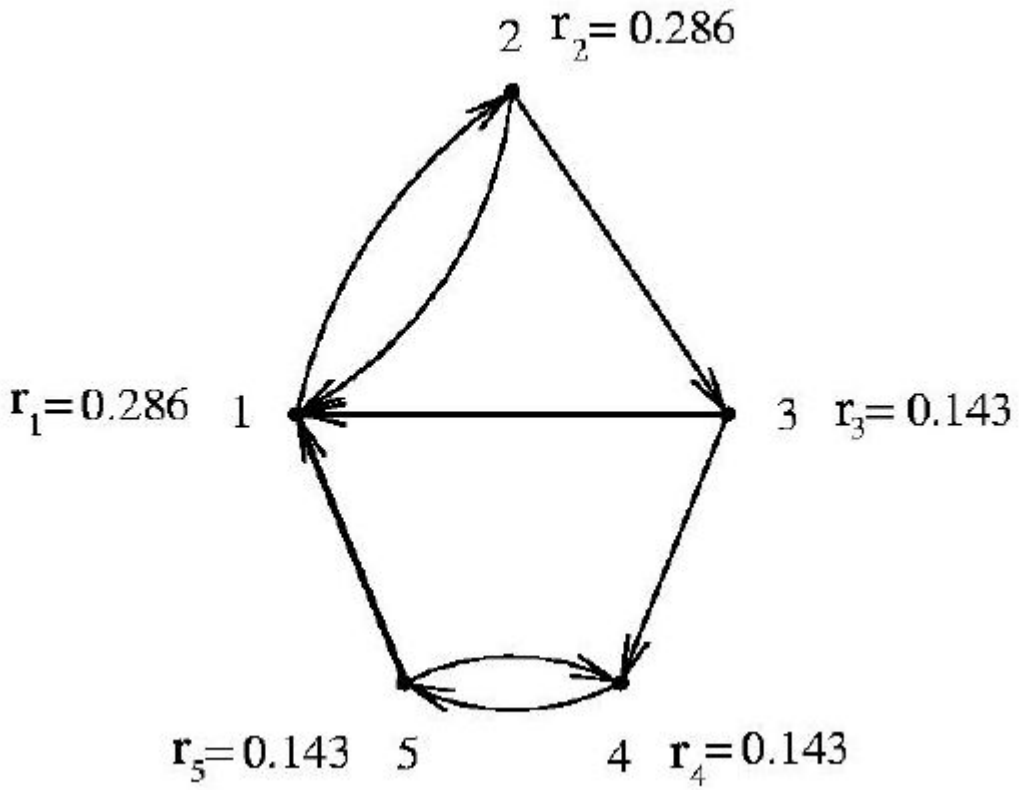
- ➔ *PR[A]* è il valore di PageRank della pagina A che vogliamo calcolare.
- ➔ *N* è il numero totale di pagine note.
- ➔ *n* è il numero di pagine che contengono almeno un link verso A. P_k rappresenta ognuna di tali pagine.
- ➔ $PR[P_k]$ sono i valori di PageRank di ogni pagina P_k.
- ➔ $C[P_k]$ sono il numero complessivo di link contenuti nella pagina che offre il link.
- ➔ *d (damping factor)* è un fattore deciso da Google e che nella documentazione originale assume valore 0,85.

Il **Damping Factor** può essere aggiustato da Google per decidere la percentuale di PageRank che deve transitare da una pagina all'altra e il valore di **PageRank minimo** attribuito ad ogni pagina in archivio.

Dalla formula si nota quindi che all'aumentare del numero di link complessivi dei siti che puntano ad A il PageRank aumenta.

Esempio: Damping Factor 1

```
web = nx.DiGraph()
web.add_edges_from([
   (1, 2),
   (2, 1),
   (2, 3),
   (3, 4),
   (3, 1),
   (4, 5),
   (5, 4),
   (5, 1),
])
print nx.pagerank(web, alpha=1)
nx.draw(web, with_labels=True)
plt.show()
```



Advanced IR

Per incrementare l'efficiacia dell"IR, diverse tecniche sono utilizzate:

- ➔ **Query Expansion**
- ➔ **Latent Semantic Indexing**
- ➔ Relevance FeedBack
- ➔ Ranking probabilistico

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 nelladisambiguazione)

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 (**distanza di Levenshtein**)
- ➔ 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)

Latent Semantic Indexing

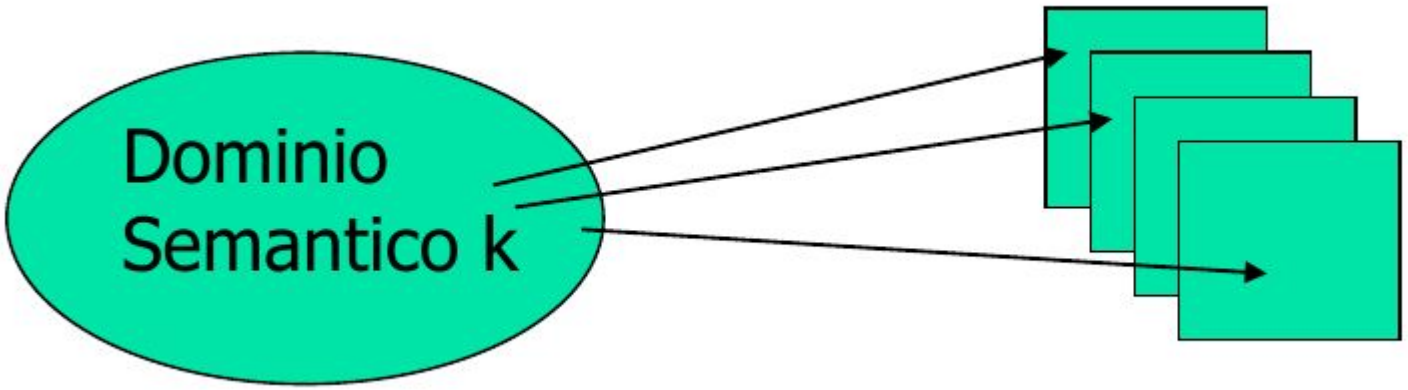
I metodi di ranking tradizionali calcolano l'attinenza di un documento ad una query sulla base della presenza o meno di parole contenute nella query: un termine o è presente o non lo è

Nel LSI la ricerca avviene per concetti: ma un concetto non è l'astrazione generalizzazione di un termine (es: golfvestiario) bensì un insieme di termini correlati (golf, maglia, vestito) detti co-occorrenze o dominio semantico

Latent Semantic Indexing

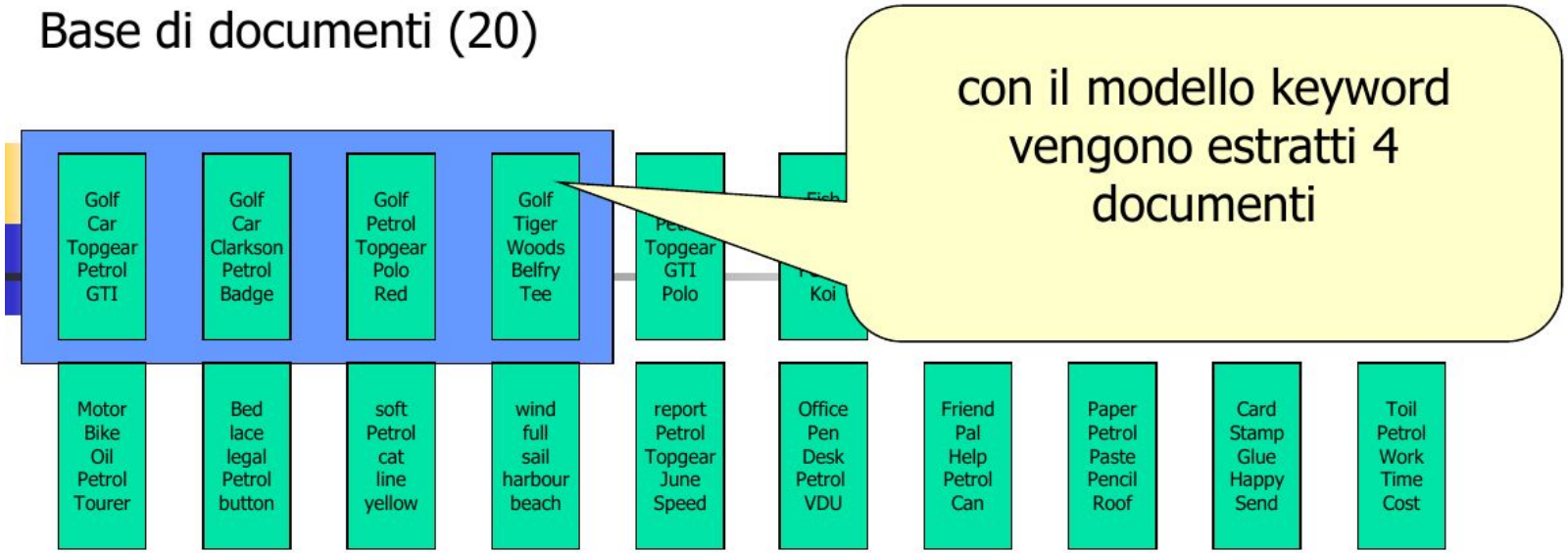
Data una collezione di documenti, LSI è in grado di rilevare che alcune n-uple di termini co-occorrono frequentemente (es: gerarchia, ordinamento e classificazione)

Se viene fatta una ricerca con gerarchia, ordinamento vengono “automaticamente” recuperati documenti che contengono anche (e eventualmente solo!) classificazione



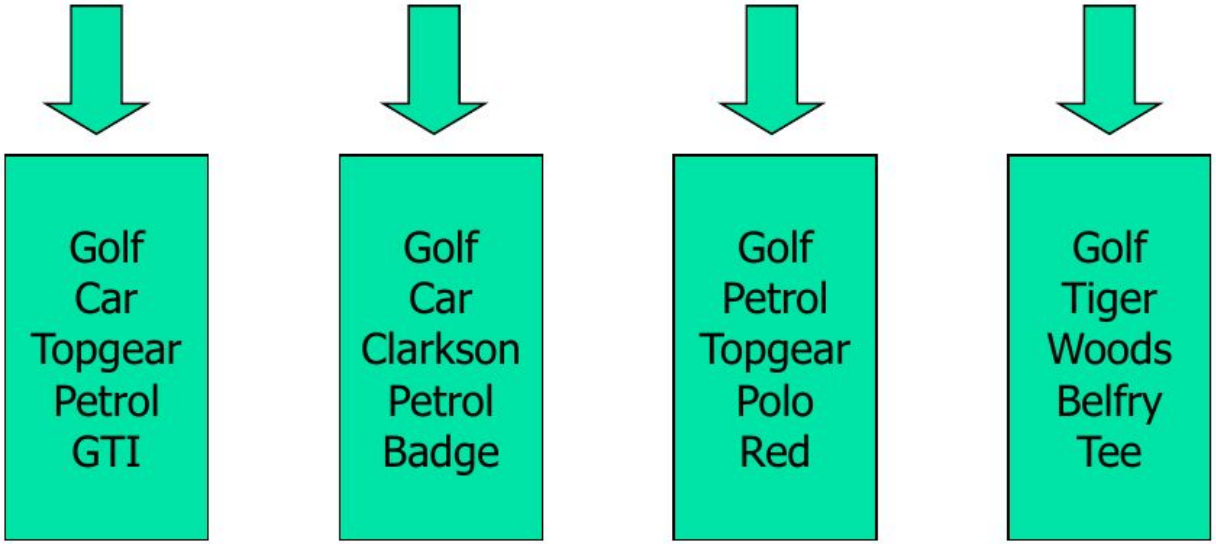
Latent Semantic Indexing

Base di documenti (20)

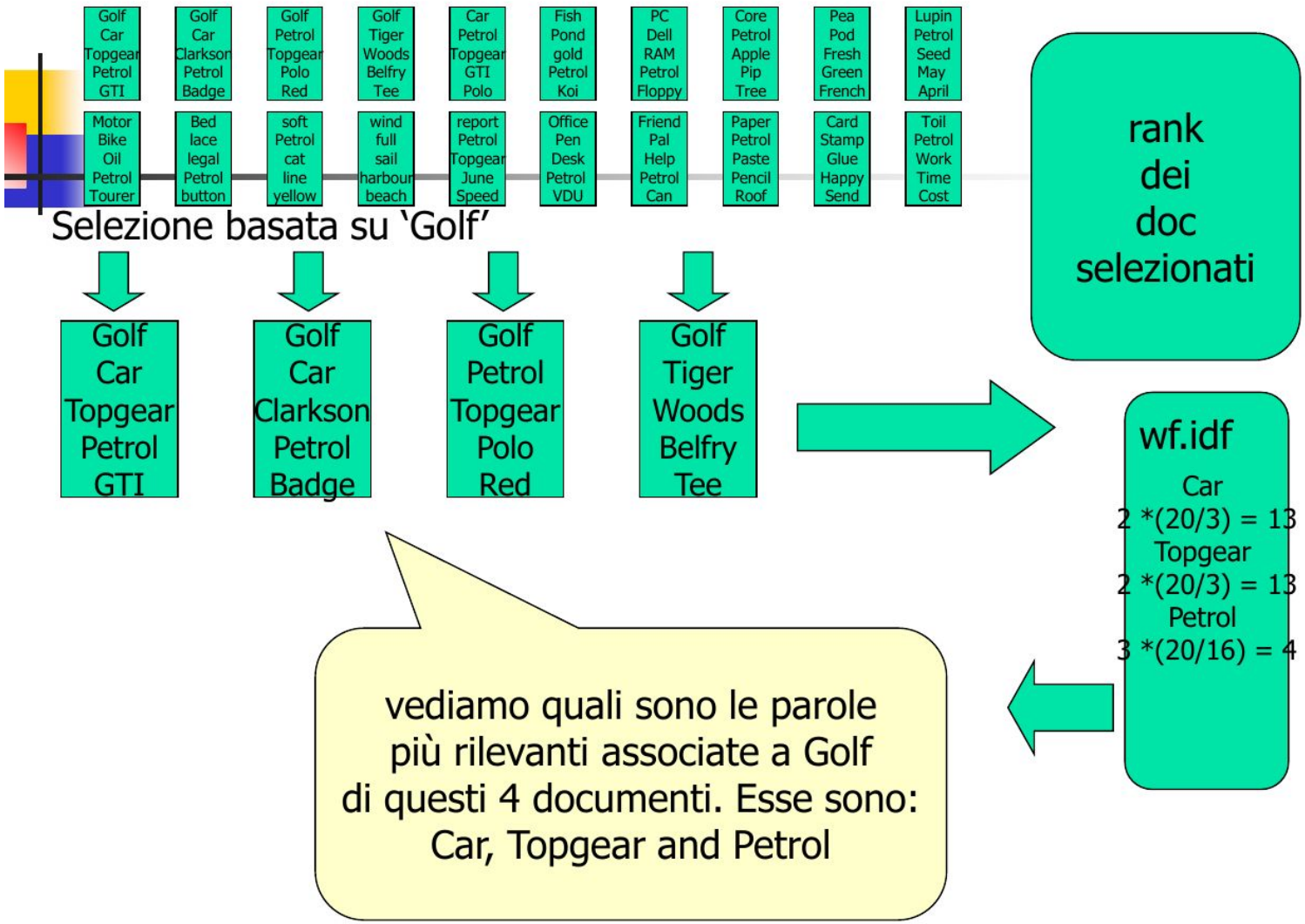


con il modello keyword vengono estratti 4 documenti

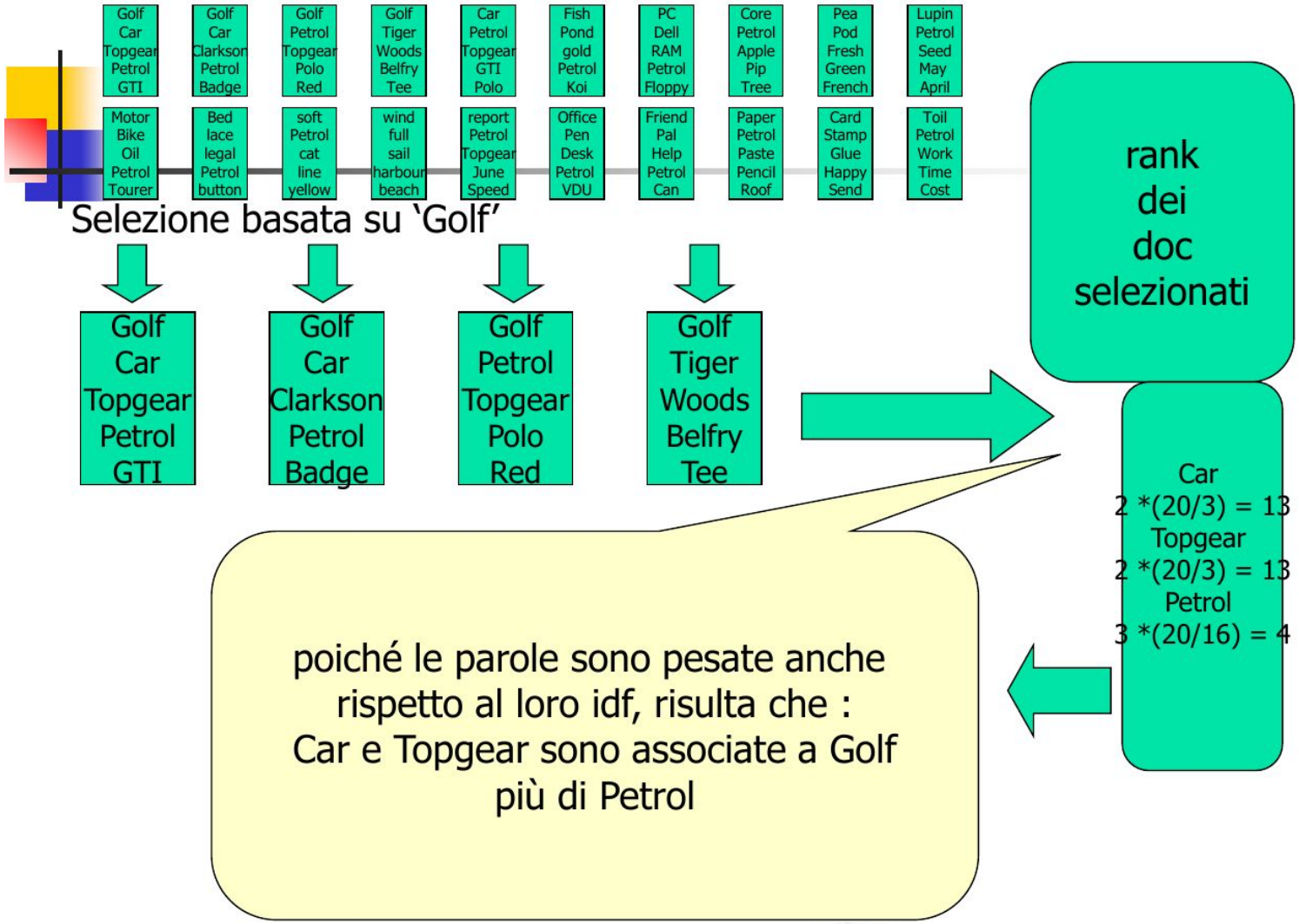
Selezione dei documenti basata sul termine 'Golf'



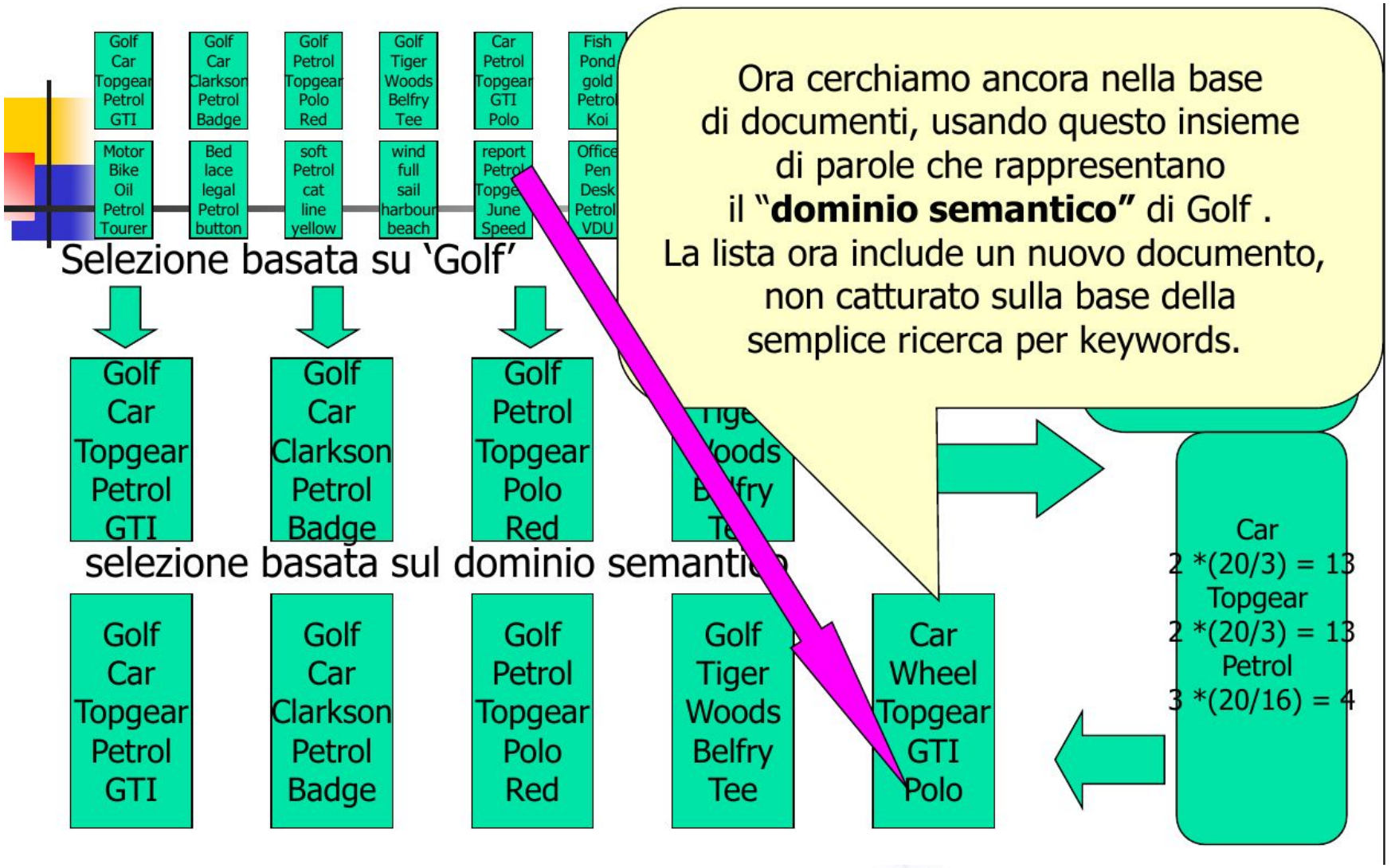
Latent Semantic Indexing



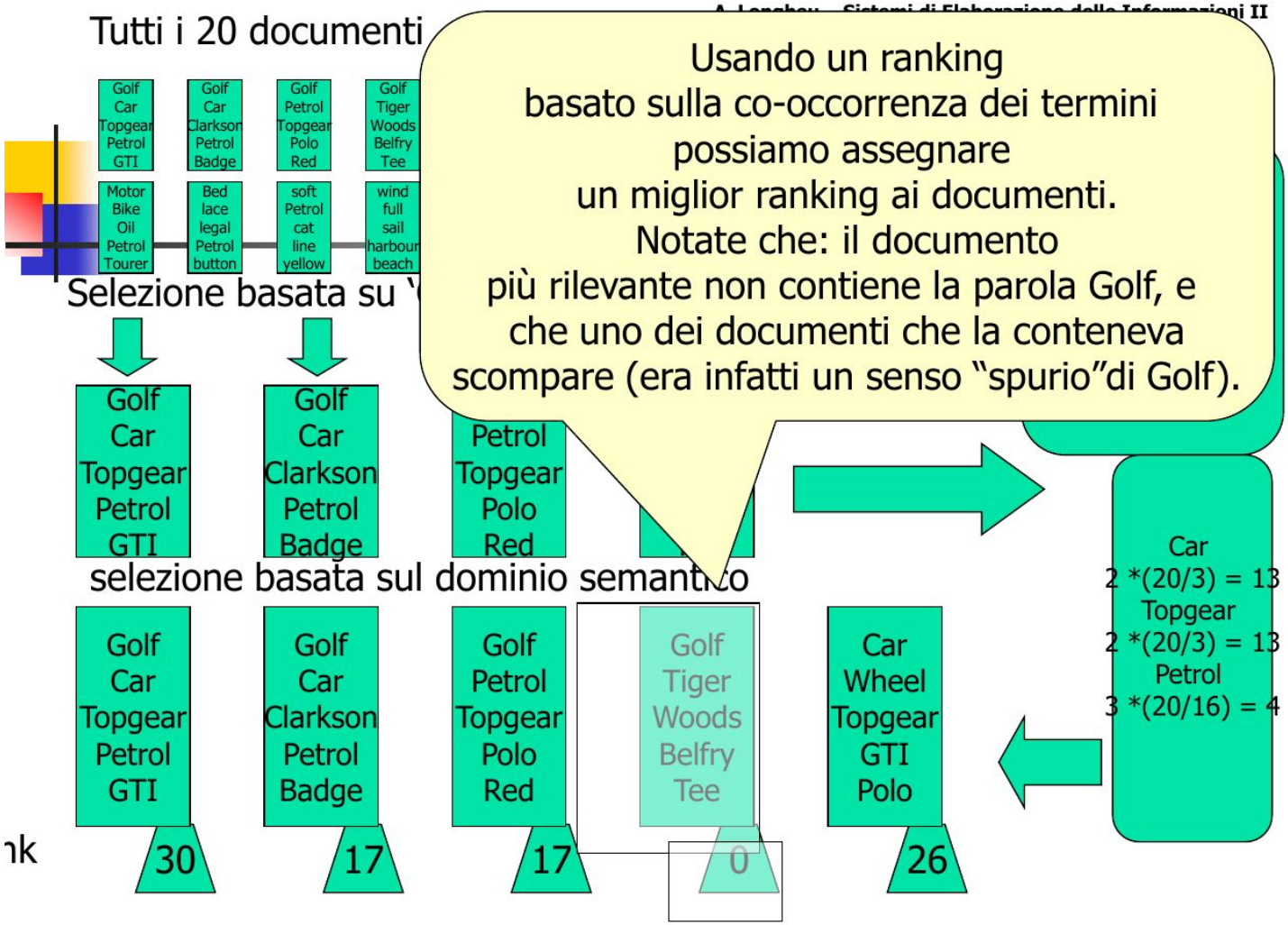
Latent Semantic Indexing



Latent Semantic Indexing



Latent Semantic Indexing



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|documentoi, 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:

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

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

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

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.

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))

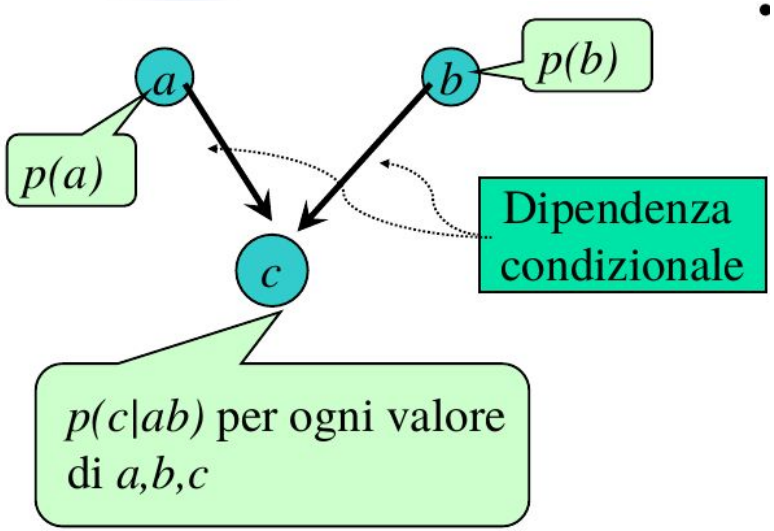
Poiché in generale è difficile stimare una certo modello probabilistico, si effettuano una serie di passaggi (ad es. invertire variabile aleatoria condizionante e condizionata con Bayes) e semplificazioni (ad es. assumere l‟indipendenza statistica di certe variabili) al fine di rappresentare il modello 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

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



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

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

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.

RelevanceFeedback

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.

RelevanceFeedback

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.

RelevanceFeedback

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

Esercitazione 4:

https://github.com/marcoortu/WAAT-2020

branch: *04-esercitazione*