Software per la ricerca di informazione

Chiara Proietti

ESTRATTO. *Questo studio è inteso a offrire una panoramica su alcune delle soluzioni proposte per il recupero dell'informazione, per riuscire a comprendere l'eterogeneità delle risposte rese al problema del* retrieval*. Si illustra l'architettura classica di un motore, proponendo il meccanismo fondante di* Google *e si prosegue con* software *in prospettiva semantico-aziendale o* web clustering*, rispettivamente con Hakia e Carrot*^{2}*.*
L'esame dei sistemi di recupero dell'informazione è scesa maggiormente nel dettaglio, affrontando uno studio di valutazione e comparazione di tali motori in termini di precisione dei risultati.

1. Il motore di ricerca: architettura e funzionamento

Strumento fondamentale dell'*information retrieval* è divenuto il motore di ricerca, un'ancora nel mare di una immensa mole di informazione. Dietro l'interfaccia utente in attesa di un'interrogazione per poterla elaborare e rispondervi, vi è un processo di acquisizione dati e indicizzazione delle risorse che costituiranno il terreno di ricerca.

È possibile ricondurre queste attività a tre fasi essenziali:

- *crawling*;
- indicizzazione;
- ricerca.

Innanzitutto il motore di ricerca inizia con l'individuare il testo da indicizzare per strutturarlo in indici che garantiscano un accesso rapido a tali risorse nel successivo recupero in fase di ricerca. Ciò implica

un'acquisizione del contenuto, mediante *crawler* o *spider*, che selezionano i dati necessari per la costruzione dell'indice. La fase di ricerca e di risposta alla *query* opererà proprio su quest'ultimo.

Per comprendere meglio come tutto ciò avviene, si scende ora nel dettaglio dell'architettura e del funzionamento di un motore di ricerca. A tal proposito, si prende in esame una classica strutturazione, su cui poggia il più famoso motore di ricerca web odierno, ossia Google.

Si attinge a quel poco reso noto, riguardo le logiche e i meccanismi di funzionamento di questo sistema di ricerca web, dai fondatori S. Brin e L. Page ancora studenti della Standford University all'epoca della pubblicazione del *paper* "*The anatomy of a large-scale hypertextual web search engine*" [1] (Figura 1).

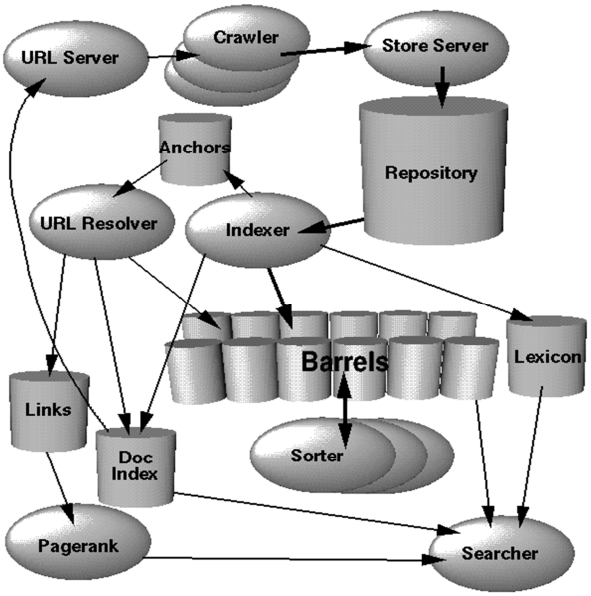


Figura 1. Architettura del motore Google.

[1] S. BRIN. e L. PAGE, "The anatomy of a large-scale hypertextual web search engine", in *Proceedings of the seventh international conference on World Wide*, Stanford, 1998, pp. 107-117 = http://infolab.stanford.edu/~backrub/google.html

I motori di ricerca risiedono su un server e avviano sulla Rete programmi, gli *spider* o *crawler*, per la ricerca automatica dei contenuti nei siti web e li copiano sul server. Da qui segue la fase di indicizzazione, che può avvenire mediante *metatag*, inseriti dal progettista della pagina web per esplicitarne i contenuti, e *full-text*, dove ogni parola del testo viene ricondotta a parola chiave per la ricerca. Ciò consente la creazione di un documento-dizionario per il motore, in cui a ogni termine viene associato il riferimento del documento in cui questo è presente, generando il cosiddetto *indice inverso*, che permette una notevole riduzioni dei tempi di ricerca, ma, al contempo, rende cruciale la selezione dei termini da inserire nell'indice.

Come già accennato, un motore di ricerca inizia a operare inviando dal server di URL una serie di programmi, *crawler*, che visitano un numero di siti web da analizzare, inclusi in una lista presente sul server.

La fase di *crawling* è un momento complesso e delicato che vede coinvolti migliaia e migliaia di server. Ogni *crawler* può tenere attive più di trecento connessioni contemporaneamente e superare le cento pagine analizzate al secondo. Tali pagine sono successivamente memorizzate e compresse nello *storeserver* in un *repository*[2], che contiene tutto il contenuto HTML delle pagine web analizzate in sequenza, con un prefisso che ne indica l'ID di documento, la lunghezza e l'URL. Alle pagine e agli URL dei siti analizzati infatti si associa un ID di documento e si prosegue all'indicizzazione grazie all'*indexer* e *sorter*.

Innanzitutto l'indice ordina gli ID dei documenti, inserisce dati di ogni occorrenza relativi allo status del documento, informazioni statistiche e di *checksum*[3] di questi, nonché un puntatore di identificazione legato all'URL e al titolo.

L'indice punta a una lista contenente solo gli URL, che a loro volta sono convertite in ID di documento da un ulteriore file di *checksum*.

---

[2] Un *repository* è un ambiente di un sistema informativo in cui vengono gestiti i metadati, attraverso tabelle relazionali; l'insieme di tabelle, regole e motori di calcolo tramite cui si gestiscono i metadati prende il nome di *metabase*. Si tratta di un ambiente che può essere implementato attraverso numerose piattaforme hardware e sistemi di gestione dei database (o DBMS, acronimo del corrispondente termine ossia *DataBase Management Systems*).

[3] Il *checksum* (tradotto letteralmente significa: somma di controllo) è una sequenza di bit che, associata al pacchetto trasmesso, viene utilizzata per verificare l'integrità di un dato o di un messaggio che può subire alterazioni durante la trasmissione su canale di comunicazione.

In esso gli URL sono elaborate dalla macchina e vi si opera una ricerca binaria per recuperare l'ID del documento. È opportuno precisare che la conversione in ID anche per gruppi di URL è operata dall'*URL resolver*, con il quale si ha un notevole risparmio di spazio su disco per la memorizzazione delle informazioni. Inoltre questo analizza i file dei link degli URL esaminati e li converte negli URL veri e propri con l'opportuna associazione dell'ID di documento. Inserendo il nome dell'URL in un ulteriore indice, si definisce una base di dati di link che permettono successivamente di calcolare il *PageRank*[4] dei documenti.

Tornando all'indice, questo, oltre ad analizzare i link interni alle pagine e a conservare le informazioni relative a essi, legge i dati nel *repository*, decomprime i documenti e li analizza, trasformandoli in un elenco di termini, noti come *hit*. Queste ultime corrispondono esattamente a una lista di occorrenze, in un documento, di una particolare parola e contengono non solo la parola, ma anche la posizione della stessa nel documento, la dimensione del suo font e la resa in maiuscolo dello stesso testo. Le modalità di registrazione sono molteplici, in quanto si cerca di raggiungere nel modo più ottimale possibile una codificazione di questi dati, cosicché possa essere efficiente per l'archiviazione in termini di spazio e per il recupero immediato dell'informazione.

È possibile distinguere due tipologie di *hit*:

- *fancy*, che includono l'URL, il titolo, l'*anchor text* o i *meta-tag*.
- *plain*, che includono tutte le altre informazioni relative ai bit di capitalizzazione, di dimensione del font e i bit che rappresentano la posizione della parola nel documento.

Le dimensioni di una lista di *hit* è definita prima dell'archiviazione delle *hit* stesse, in quanto ogni lista è combinata con un ID di parola dell'indice precedente e da un ID di documento dell'indice inverso, limitandolo così tra i 5 e gli 8 bit.

L'indice precedentemente costituito con gli ID di parola è memorizzato e ridistribuito in *barrel* di indice distinti per tipologia.

[4] Il *PageRank* è un algoritmo di analisi che assegna un peso numerico a ogni elemento di un collegamento ipertestuale di un insieme di documenti, e dunque nel World Wide Web, con lo scopo di quantificare la sua importanza relativa all'insieme di documenti. L'algoritmo può essere applicato a tutti gli insiemi di oggetti collegati da citazioni e riferimenti reciproci. Brevettato dalla Stanford University, è oggi un marchio Google.

Ognuno di essi contiene infatti un numero di ID di parola, l'ID del documento che le contiene e una lista di *hit* che corrisponde a tali termini registrati dagli identificativi. L'indice inverso consiste degli stessi *barrel*, ma interagisce con il *sorter*. Il *sorter* attinge dai *barrel*, distinti per ID di documento e li ridefinisce per ID di parola per creare un indice inverso.

Un programma, noto come DumpLexicon, associa la lista di ID di parola con il lessico prodotto dall'indice e rigenera una nuova lista di termini che possono essere utilizzati dal *searcher*.

La terza fase, che segue appunto quella di *crawling* e di indice, è quella di *ricerca*. Questa fase, altrettanto delicata, è fondamentale per il recupero di un'informazione qualitativamente valida rispetto all'interrogazione immessa. Diviene particolarmente complesso riuscire a considerare tutti i fattori relativi al termine inserito per la ricerca, registrati nelle *hit*; ancor più gravoso diviene quando si ha una ricerca per più parole-chiave, prima esaminate singolarmente e infine valutate per la loro prossimità/similarità complessiva con le *hit* del documento.

In ogni modo la funzione di *ranking*, ossia di ordinamento e classificazione dei risultati, prende in considerazione molti parametri, a cui attribuisce peso diverso. Possono o meno essere resi noti all'utente in base al grado di trasparenza del sistema in questione.

A seguito della fase di ricerca, il motore non arresta la sua attività, ma generalmente sfrutta il feedback dell'utente relativamente al risultato offerto. Ciò andrà a impattare con la funzione di *ranking*, ridefinendo, in successive ricerche, l'ordine dei risultati in base alla percezione di qualità rilevata direttamente dall'utente.

Sul web server, utilizzando questa lista, l'indice inverso e l'algoritmo di *PageRank*, si elabora la risposta all'interrogazione proposta al motore.

L'architettura di un motore è complessa, ma è altresì tanto variegata, considerando che, pur poggiandosi su meccanismi e logiche architetturali che poco si discostano da quelli appena presentati, molti prodotti per il *retrieval* sono in continua ricerca di soluzioni differenti per poter offrire risultati sempre migliori in termini di qualità e appropriatezza rispetto alla richiesta dell'utente.

Si affrontano di seguito due soluzioni che nascono da un approccio semantico e aziendale e un approccio *web clustering* al recupero dell'informazione in Rete.

2. Hakia: un motore semantico

Il motore di ricerca Hakia è un interessante caso di studio, in quanto include in sé le caratteristiche di un motore di ricerca semantico e aziendale. Si configura in una struttura modulare, che oltrepassa i limiti del *matching* statistico delle parole-chiave della *query* nel documento e offre una particolare soluzione al problema della *scalabilità*[5] del Web. Sovverte il classico meccanismo dell'indice inverso e arricchisce la propria ricerca di un raffinamento semantico dei risultati.

Il metodo QDEX, su cui il motore in esame poggia, sfrutta un processo inferenziale inverso rispetto a quello tradizionale ed elabora, partendo dalle frasi individuate nel testo, una serie di domande di cui queste possono essere la risposta. Si configura una sorta di nodo di risposta, frutto di una comparazione tra i contenuti simili individuati per quella stessa porzione di testo in pagine web differenti, per individuare quello che si configura come risultato migliore alla domanda derivata.

Tale algoritmo permette dunque di identificare il binomio "domanda-risposta" preventivamente all'azione dell'utente e di memorizzarlo in un indice più ristretto, su cui l'operare sarà più immediato. Tra i vantaggi, la possibilità di attivare una ricerca distribuita, flessibile nell'archiviazione e migliore nella gestione della scalabilità.

Hakia coniuga la propria indicizzazione QDEX e la struttura ontologica, abbandonando algoritmi basati sulla popolarità per il *ranking* e tentando di offrire una risposta entro una specifica sezione di testo e non di un intero documento. Tra i suoi obbiettivi c'è certamente quello di migliorare la precisione, l'ampiezza della varietà lessicale trattata, accrescendo i propri indici non linearmente con la crescita del Web, bensì con un incremento dei soli nuovi contenuti da verificare secondo

[5] Si intende la capacità di un sistema di rispondere a un determinato numero di ricerche in termini di secondi mediante un indice che copre un certo numero di pagine web, che crescono in maniera proporzionale alla crescita dei contenuti in Rete.

criteri di rilevanza e credibilità, per evitare le eventuali ripetizioni di dati. Mira inoltre a indici dalle quantità di dati ridotte che possano consentire una risposta rilevante in tempi minimi per l'utente. Questi registrano l'ID del paragrafo dove il testo analizzato è reperibile. Tali file di indicizzazione sono conservati in una serie di server mediante una codificazione in *hash* che semplifica e rende immediata ogni operazione su di esso.

Il processo, generando domande e interrogazioni consone alla frase o termine analizzato, comparerà queste, anche sulla base dei valori percentuali statistici di occorrenza delle differenti forme, e provvederà alla conservazione. La generazione di queste varianti linguistiche, nota come *breeding*, segue lo stesso processo di apprendimento umano, che, al seguire della comprensione del significato, memorizza una serie di conoscenze associate, che in questo caso divengono nodi focali di risposta per l'elaborazione dei risultati di una *query*.

Questa fase integra un approccio *fuzzy*[6], che utilizza il metodo "*Bag-of-Words*"[7] per l'identificazione del significato delle parole immesse per la ricerca e opera una cancellazione delle sequenze che non risultano realmente utilizzate. Alla base di tutto ciò, ad accrescere la componente semantica della ricerca del motore in questione, anche il coinvolgimento di un ontologia, detta *OntoSem*, che permette di valutare e validare le sequenze create dall'algoritmo e di mutare le sequenze in differenti forme linguistiche.

Inoltre un modulo indipendente di cui si compone questo motore per la ricerca è il *Semantic Rank*. Aggiunge alla ricerca l'elaborazione dei dati in una dimensione sintattica, ontologica e morfologica, che restringe il *retrieval* nell'ambito di un paragrafo e non dell'intero documento. In tale prospettiva è bene sottolineare come Hakia prenda in considerazione nella ricerca varianti morfologiche, sinonimi, iperonimi e iponimi, *query* in linguaggio naturale, formati speciali, inferendo su concetti e conoscenze.

---

[6] La logica *fuzzy* o logica *sfumata* o logica *sfocata* è una logica in cui si può attribuire a ciascuna proposizione un grado di verità compreso tra 0 e 1. È una logica polivalente, e pertanto un'estensione della logica booleana.

[7] Il modello *Bag-of-Words* non tiene conto dell'ordine, grammaticale o sintattico, dei termini in un documento, ma considera quest'ultimo una semplice sacca di parole da cui estrarre tali termini e valutarne l'occorrenza degli stessi quantitativamente.

Quanto soluzione per l'azienda, i suoi grandi pregi risiedono nell'ampia possibilità di personalizzazione; in particolare, nell'opportunità di incorporare conoscenze e linguaggi specifici dell'organizzazione, di ridefinire la pagina dei risultati, di poter avere dati statistici relativi agli stessi e di avere un supporto semantico nella ricerca che eleva il grado di precisione dei risultati. Fornisce alle aziende supporti ulteriori mediante ricerche e un'azione di ricerca distribuita, che recupera l'informazione su specifici domini web selezionati, con un aggiornamento costante delle proprie fonti che offrono dati costantemente aggiornati, assieme a meccanismi di *alerting* e avviso.

In Hakia si coglie tutto il desiderio della ricerca tecnologica attuale di operare semanticamente la ricerca, di oltrepassare i limiti della coincidenza lessicale tra lemmi di *query* e quelli nel testo recuperato, per offrire un risultato qualitativamente migliore rispetto all'esigenze dell'utente.

3. Carrot^2: un motore *web clustering*

Si presenta qui una soluzione nella prospettiva di *web clustering search* per offrire una panoramica più ampia delle proposte per un *retrieval* dell'informazione che sia sempre più adeguato a rispondere alle crescenti e innumerevoli richieste dell'utente. Si analizza a tal proposito Carrot^2, un motore di ricerca *clustering open-source* in Java, che potrebbe anche integrarsi con le altre soluzioni fin qui indagate.

La prima versione fu implementata nel 2001 dal polacco Dawid Weiss. Nel 2003 vide l'implementazione di ulteriori algoritmi di *clustering*, tra cui Lingo, pensato propriamente per il *clustering* di materiale testuale. La sua architettura è riconducibile alle classiche quattro componenti di un motore *web clustering* che rispondono alle quattro fasi essenziali, di acquisizione dei risultati della ricerca, elaborazione *ex ante* dell'input, costruzione del *cluster* e visualizzazione dei risultati.

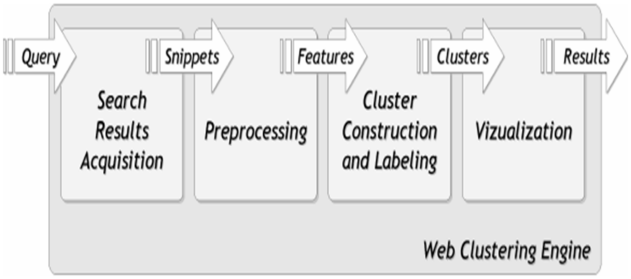


Figura 2. Processo di un *web cluster search engine*.

Nella prima fase di acquisizione dei risultati della ricerca si sfruttano generalmente tra i 50 e i 500 risultati offerti da motori di ricerca web, considerando il titolo, l'estratto contestuale che lo accompagna e l'URL, cui il risultato della ricerca rinvia. L'acquisizione può avvenire mediante le interfacce applicative di programmazione (API) di tali motori, sebbene esistano una serie di restrizioni d'accesso a queste. Altrimenti può adoperarsi estraendo i risultati dal motore, mediante l'*HTML scraping*, ossia utilizzando i dati del linguaggio di marcatura alla base della pagina web. In ogni modo, quest'ultima opzione ha notevoli risvolti negativi, a causa dei continui cambiamenti che possono presentarsi nel codice HTML e nella manutenzione che deve generalmente effettuarsi manualmente, richiedendo dispendiose risorse.

Altra alternativa, interessante in particolar modo per quel che riguarda le imprese e la ricerca da effettuarsi per contenuti di specifici domini, è l'acquisizione per indice di documento.

La seconda fase del processo si configura per convertire l'input in ingresso in una serie di caratteristiche, al fine di renderlo trattabile per l'algoritmo. Si procede, innanzitutto, all'identificazione della lingua in esame, necessaria per poter procedere a seguire con l'elaborazione dei *token*, delle entità linguistiche irrilevanti e infine selezionare i tratti linguistici d'interesse per il contesto d'interrogazione in esame.

La fase di *tokenization* permette di individuare le singole unità linguistiche che saranno poi ricondotte alle loro radici. Queste non sono altro che la porzione del termine che si fa portatrice linguistica del significato della parola e cui è possibile ricondurre una molteplicità di termini. In tal modo si estraggono le parole chiave dal testo, eliminan-

do prefissi e suffissi dal ruolo marginale per focalizzare solo sui significati essenziali nel testo.

La terza fase è quella di costruzione del *cluster* e delle *label* corrispondenti. È questa il risultato dell'elaborazione dei tratti estratti nelle fasi precedenti da parte degli algoritmi. Come è possibile comprendere, ha un notevole valore l'esatta e qualitativamente valida scelta dei termini di categoria, per poter comprenderne chiaramente i contenuti e per determinare la classificazione in base a un criterio di rilevanza.

La fase finale di visualizzazione differisce tra i diversi motori *cluster*. Si preferisce solitamente la visualizzazione ad albero, poiché compatta e completa; non mancano però versioni grafiche che mettono in maggior evidenza le relazioni di dimensioni, genere e distanza tra *cluster*, che generalmente si sfruttano per l'elevata flessibilità nella navigazione dei sottotemi.

4. Un caso di studio: Google-Hakia-Carrot

A questo punto dell'analisi offerta su alcune soluzioni pensate per rispondere a un'estrazione dell'informazione che sia semantica, ovvero adeguata alle esigenze di un'azienda, si ritiene opportuno scendere nel dettaglio di uno studio mirato e puntuale per cogliere ciò che può definirsi un buon motore di ricerca e, in particolare, qual è il valore aggiunto dell'approccio semantico alla ricerca e, ancora, se se ne può cogliere realmente tale apporto positivo.

Si affronta un caso di studio che pone a confronto tre motori di ricerca che operano in maniera differente il recupero dell'informazione: Google, Hakia e Carrot. Dal colosso della ricerca Google si muove a una comparazione con un motore aziendale e semantico come Hakia, per concludere il confronto con Carrot, un *clustering search engine*.

L'esame proposto permette di superare la prospettiva iniziale con cui si è intrapresa l'indagine, in quanto mediante la rilevazione del grado di precisione offerta nella risposta alla *query*, si cerca di comprendere più nel dettaglio anche il modo di operare dei tre motori, verificando la robustezza del sistema rispetto alla molteplice variazione dell'interrogazione.

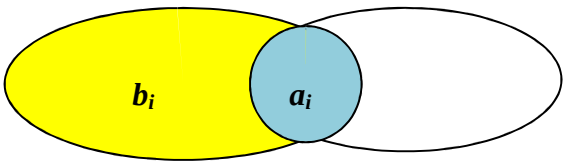
4.1. Obiettivi e modalità

Il punto di partenza di questa indagine può riassumersi in una domanda: «Quando un motore di ricerca può definirsi effettivamente utile»? La risposta corretta a tale quesito è forse la più semplice, che ogni utente è in grado di dare: la sua utilità è la qualità del risultato offerto all'interrogazione immessa nel sistema. Quanto più alta è la percentuale di significazione dei risultati rispetto al fabbisogno informativo, tanto l'utente è stato spinto a interagire con il motore.

Si può asserire che la valutazione di utilità del sistema può ricondursi alla *precision* di questo, ossia alla capacità di un sistema di rendere solo i testi rilevanti per l'interrogazione, eliminando correttamente dai risultati di ricerca quelli non utili, irrilevanti.

L'obiettivo dell'analisi che qui verrà a delinearsi è quello appunto di esaminare il grado di precisione della risposta dei motori indagati. La formula per il calcolo della precisione è generalmente $\alpha_i / \alpha_i + b_i$ dove α_i corrisponde al numero di documenti recuperati e rilevanti, b_i al numero dei falsi positivi, ossia i documenti recuperati ma irrilevanti.

Totalità dei dati resi: $a_i + b_i$



- *Risultati rilevanti*: porzione della totalità dei risultati resi e dei falsi positivi
- *Falsi positivi* : risultati irrilevanti

La rilevanza o, meglio, *attinenza*, come raccomanda Mizzaro[8], è un concetto sì fondamentale nel settore ***dell'****information retrieval*, ma altrettanto complesso nella sua definizione, che ha visto negli anni susseguirsi varianti su varianti.

[8] S. MIZZARO, "Le differenti *relevance* in *information retrieval*: una classificazione", in *Proceedings of the AICA Annual Conference*, Cagliari 1995, vol. I, pp. 361-368 (per il testo in italiano si vd. http://users.dimi.uniud.it/~stefano.mizzaro/research/ papers/rel-aica.pdf)

Per esemplificare il grado di vaghezza dei contorni di tale definizione, si propone dapprima una definizione che negli anni settanta cercò di raccogliere in sé le diverse asserzioni fino ad allora compiute, relativamente a questo termine. Saracevic afferma che:

> l'attinenza è la (A) di un (B) esistente tra un (C) e un (D) come determinato da un (E)[9]

dove si rappresenta con :

A: misura, stima, giudizio;
B: utilità, coincidenza, soddisfazione;
C: documento, informazione resa;
D: interrogazione, fabbisogno informativo;
E: utente- richiedente, esperto.

Questo tentativo di formalizzazione della definizione è proseguito nel tempo e resta una delle sfide maggiori ancora oggi in questo ambito di studi. È bene però ricordare almeno altre due definizioni di rilevanza per comprendere meglio l'analisi qui proposta: la definizione di Cuadra e Katter[10] e quella di Cooper.

Nel primo caso si parla di rilevanza come della corrispondenza nel contesto tra un'interrogazione per l'informazione e il materiale appropriato all'espressione della richiesta formulata[11]. Nel secondo caso è definita come la pertinenza, in termini di implicazione logica, dove si considera l'utilità di quanto restituito dal sistema di recupero relativamente alla relazione con il tema, alla qualità, credibilità, importanza e numerosi altri fattori.

Se l'impossibilità di formulare un criterio di rilevanza netto e ben definito può inizialmente indurre a ritenere l'analisi che segue poco oggettiva, è bene considerare che lo scopo effettivo di un motore di ricerca è soddisfare la sua utenza.

---

[9] A.M. REES e T. SARACEVIC, "The measurability of relevance", in *Proceedings of the American Documentation Institute*, 3, 1966, pp. 225-234. Si vd. anche S. MIZZARO, "Le differenti *relevance* in *information retrieval*: una classificazione.", cit.

[10] S. MIZZARO, "Relevance: the whole history" in *Journal of the American society for information science*, vol. 48, no. 9, 1997, pp. 810-832.

[11] *ibidem*.

Un'utenza che, come asserisce Mizzaro, si avvicina a tale interazione per rispondere a un bisogno informativo implicito, originato da un problema o da uno scopo, è la percezione stessa del proprio scopo/problema che l'utente possiede. Da una fase profonda e interiorizzata, segue una fase di esplicitazione di un fabbisogno informativo che spesso lo stesso utente non comprende nella propria totalità, in quanto spinto proprio a colmare una lacuna conoscitiva in un settore non padroneggiato egregiamente.

Anche per questo, nell'analisi si è scelto di operare una serie di interrogazioni che subiranno delle variazioni linguistiche per poter verificare sì come il sistema vi risponde, ma anche per considerare e simulare delle reali condizioni operative rispondenti a diversi gradi di esplicitazione del fabbisogno informativo.

Si procede da un termine chiave di ricerca che esplicita il concetto di partenza dell'interrogazione, per poi utilizzare le sue varianti definitorie, tratte dal dizionario[12]. Si lavora sulla definizione del lemma utilizzandola al posto del termine chiave e integrandola talvolta con questo o con un termine di disturbo, valutando il diverso grado di rilevanza dei documenti di risposta offerti dai tre sistemi di *retrieval* al mutare dell'interrogazione.

Innanzitutto le quattro *query* utilizzate si basano sui seguenti termini chiave: *accessibility*[13], *usability*[14], *interface*[15], *interactivity*[16].

---

[12] Le definizioni sono state tratte dal dizionario monolingua della lingua inglese: *English Dictionary for advanced learners*, MacMillan Education, Londra 2012.

[13] La proprietà dei sistemi informatici di essere fruibili senza discriminazioni derivanti da "disabilità". Per "disabilità" si intende qualsiasi restrizione o impedimento del normale svolgimento di un'attività derivante da una menomazione fisica o cognitiva. Vd. "Accessibilità", 2002, http://www.pubbliaccesso.gov.it/biblioteca/quaderni/rif_tecnici/quaderno_4.doc

[14] «L'insieme delle caratteristiche architetturali delle interfacce *software* uomo-macchina, che consentono all'utente di interagire con soddisfazione rispetto agli scopi definiti dagli specifici programmi»: cfr. E. ZUANELLI, *Manuale di linguaggio, comunicazione e applicazioni digitali*, Colombo, Roma 2006.

[15] L'interfaccia è un punto d'incontro tra due diverse entità che si suppone vengano in contatto e vengano ravvicinate per comunicare. Ciò implica che l'interfaccia condivide con il segno la sua natura semiotica. Vd. M. NADIN, "Interface design: A semiotic paradigm", in *Semiotica* 69, Amsterdam 1988, pp. 269-302.

[16] La caratteristica di un prodotto digitale finalizzato a rendere esecutiva l'interazione uomo-computer. In tale contesto il sistema di azione-interazione uomo-macchina sarà l'insieme di eventi interattivi informatici volti al dialogo dell'utente con l'applicazione al fine di svolgere azioni per precisi scopi. A tale esigenze la macchina risponderà in modo adeguato alle

I termini scelti sono stati pesati sulla base del grado di tecnicità che posseggono, sebbene questo non sia eccessivamente marcato, per evitare che termini troppo vaghi rendano impossibile l'indagine. Hanno un minimo grado di polisemia, che interessa nell'esame in atto per verificare il comportamento del motore nel trattamento della possibile ambiguità lessicale rispetto alla specificità del dominio di appartenenza.

Le varianti delle quattro *query* espresse dal termine chiave esplicito sono otto:

- la parola chiave e la definizione dello stesso lemma;
- la sola definizione senza la parola chiave;
- la parola chiave, la definizione e una parola di disturbo non attinente in alcun modo al dominio del termine di ricerca;
- la definizione e una parola di disturbo;
- la parola chiave e le parole ritenute rilevanti nella definizione estesa;
- le sole parole ritenute rilevanti nella definizione estesa;
- la parola chiave, le parole ritenute rilevanti e la parola di disturbo;
- le parole ritenute rilevanti e la parola di disturbo.

L'obiettivo è dunque un'analisi prestazionale dei tre motori sulla base di una ricerca che desidera risposte all'interrogazione di carattere definitorio. Quanto più il risultato offrirà una definizione accurata rispetto all'interrogazione, tanto più sarà considerato rilevante.

Si tenta di valutare la robustezza del sistema, ossia se questo riesce a cogliere lo stesso grado di significazione nonostante la variante linguistica adoperata, altrimenti in che grado si verifica il cambiamento nell'offerta dei risultati e il livello di instabilità riscontrata. Il mutare del grado di precisione dei risultati ottenuti dall'interrogazione consente di comprendere l'orientamento di tali motori, siano questi più incentrati sull'uso di termini chiave o su *query* più estese, tendano a logiche puramente statistiche oppure semantiche.

---

aspettative dell'utente eseguendone le richieste e gli ordini. Vd. E. ZUANELLI, *Manuale di linguaggio, comunicazione e applicazioni digitali*, cit.

Si propone un esempio utilizzato nell'interrogazione per meglio comprendere come si sono scelti gli elementi di *query* e con quali obiettivi.

Termine di query	Variante di query
Keyword Only (KO)	***Interactivity***
Keyword + Definition (K+Def):	***interactivity + involving people communicating with each other and reacting to each other***
Definition Only (DefOnly):	***involving people communicating with each other and reacting to each other***
Keyword + Random Word + Definition (K+RandW+Def):	***interactivity + bacon + involving people communicating with each other and reacting to each other***
Random Word + Definition (RandW+Def):	***bacon + involving people communicating with each other and reacting to each other***
Keyword + Relevant words (K+RelWs):	*involving people* ***communicating*** *with each other and* ***reacting to each other***
Relevant words Only **(RelWsOnly):**	*involving people* ***communicating*** *with each other and* ***reacting to each other***
Keyword + Random Word + Relevant words (K+RandW+RelWs):	***interactivity + bacon*** *+ involving people* ***communicating*** *with each other and* ***reacting to each other***
Random Word + Relevant words (RandW+RelWs):	***bacon*** *+ involving people* ***communicating*** *with each other and* ***reacting to each other***

Tabella 1. Termini e varianti di *query*.

Le varianti sono dunque lo strumento essenziale utilizzato per perturbare il sistema e verificare come, variando in povertà espressiva e in grado di rumorosità, con quale livello di robustezza questo risponde.

Il punto di partenza è un termine specifico immesso per la ricerca di una definizione o di un documento che la contenga. È bene precisare che negli obiettivi posti a premessa dell'esame, la ricerca di una definizione del concetto espresso dal termine chiave costituisce il criterio di valutazione del grado di rilevanza del documento risultante[17].

[17] I risultati della rilevazione sono disponibili in Appendice.

L'uso della sola parola chiave tecnica implica un alto livello di povertà dello stimolo reso al motore, ma al contempo ridotto nella sua rumorosità. Segue un'esplicitazione più esaustiva del concetto, utilizzando la definizione dello specifico lemma, proprio come farebbe un utente che tenta di esplicitare un concetto che non conosce appieno.

In tal modo si valuta in più anche la gestione di un'interrogazione composta da più termini che consente di svelare il grado in cui il motore poggia su un meccanismo statistico. Si tenta inoltre mediante l'integrazione del termine chiave con la definizione di aumentare il grado di rumore.

Si muove verso una dimensione più propriamente semantica nell'indagine, operando senza il termine chiave iniziale e lasciando la sola definizione di questo, che dovrebbe ricondurre al concetto iniziale stesso. Si potrà riscontrare così se il motore segue un approccio semantico per la ricerca oppure se ha una sola valenza statistica. In quest'ultimo caso il motore si perderà in documenti con i soli termini inseriti e non su contenuti propriamente attinenti al termine chiave implicitamente ricercato.

Per la verifica del grado in cui il motore opera semanticamente, si inserisce successivamente nell'indagine una variante che non contiene l'intera definizione del termine, bensì solo un numero ristretto di parole contenute in questa che possano essere considerate l'essenza della definizione stessa, esprimendo senza *stop-word* o termini poco rilevanti il concetto indagato. Se ne valuterà la combinazione o meno del termine chiave iniziale, valutando così una combinazione meno ricca della definizione e quindi la gestione di un'interrogazione con un minor tasso di rumorosità.

Queste varianti di *query* presentate sinora sono state ulteriormente manipolate per accrescere il grado di difficoltà della ricerca per il sistema di *retrieval* d'interesse. Si è inserita infatti una variante, per ognuna di quelle finora presentate, contenente un termine di disturbo, che causa un maggior grado di rumore per il motore stesso nella sua attività di recupero.

Il termine scelto in questo caso è *bacon*, un termine nettamente riconducibile al dominio alimentare/culinario, estremamente distante da quello più tecnico-informatico e di comunicazione digitale dei termini scelti per le interrogazioni. Si cerca così di verificare come i diversi

sistemi affrontino il diverso grado di complessità e come si mantengano stabili nel rendere risultati con un accettabile livello di precisione.

Si rende di seguito esplicita la costruzione del metodo d'indagine utilizzato per evidenziare la logica a questo sottesa. La resa grafica della metodologia di costruzione delle varianti di *query* evidenzia quanto generalmente si attende da un sistema di *retrieval*, un decrescere in prestazioni con l'aumentare del fattore rumore e della ricchezza espressiva (Figura 3).

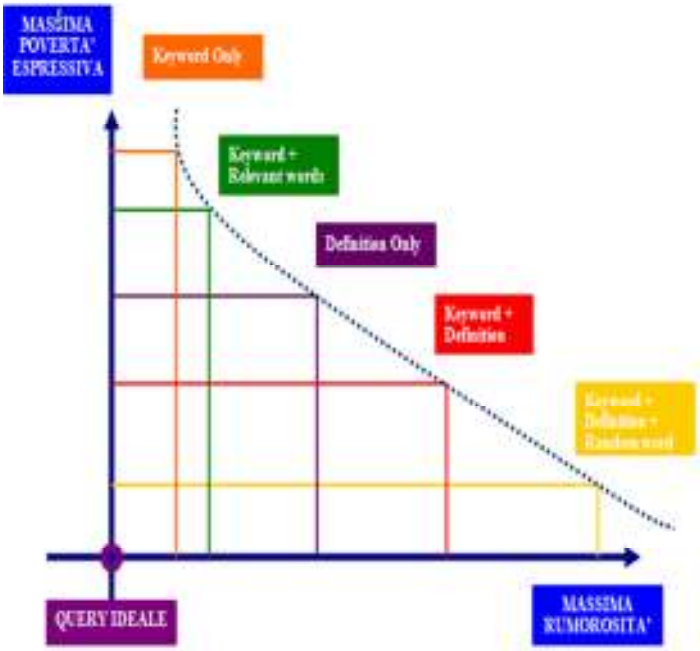


Figura 3. Metodologia di costruzione delle varianti di *query*.

4.2. Criteri di valutazione

La valutazione dei sistemi tiene conto, nel caso di studio qui presentato, dei primi dieci risultati offerti, in quanto si tratta in genere del numero di risultati nella prima pagina di risposta alla *query*. Ciò in considerazione della necessità di fornire il risultato più rilevante

all'interrogazione nel minor tempo possibile, richiedendo all'utente il minor sforzo cognitivo. Ciò permette nell'esame di valutare pertanto il diverso livello di efficienza mostrato dai motori comparati.

Si opera così una valutazione per ogni documento reso dai tre motori secondo tre livelli:

1 = irrilevante/ poco rilevante
2 = mediamente rilevante
3 = altamente rilevante

La rilevanza, come già asserito, è valutata sulla base del risultato e della sua offerta di un contenuto definitorio del termine inserito. Si calcola nell'esame il numero di documenti che sono dal mediamente all'altamente rilevanti (Vd. Appendice, Caso I). Al contempo si verificano anche le risposte dei soli valori più elevati di rilevanza solo con valore 3, (Vd. Appendice, Caso II), per un riscontro del motore che offre il maggior grado di risposte efficienti e soddisfacenti per l'interesse mostrato nella *query*. Si procede così per le quattro query e le rispettive varianti, stabilendo una media delle percentuali di precisione dei tre motori.

4.3. Risultati

Per valutare i risultati si propongono una serie di grafici che permettono di valutare il diverso andamento dei tre motori rispetto alla curva tradizionale che vede crescere proporzionalmente la rumorosità con il crescere del numero delle parole immesse e che a sua volta è inversamente proporzionale alla povertà dello stimolo immesso.

La condizione ideale, cui dovrebbero tendere i motori, è una regolarità del trattamento, anche nel caso di rumorosità elevata. La *query* ideale è infatti la *query* che mostra, alla massima ricchezza espressiva, la minima rumorosità.

Si assiste dunque a comportamenti molto differenti. Google si mostra molto consistente, sebbene incida molto nella sua precisione l'accrescere della rumorosità. Si veda come i valori tratti dalla sola parola chiave (KO) siano i più elevati in assoluto, ma come crollino al crescere del rumore e del numero dei termini di *query*, anche se il

reinserimento della parola chiave con la definizione (k + Def) genera un picco di notevole interesse.

Nel caso di Carrot è estremamente interessante il riscontro di valori più ridotti rispetto a Google, ma al contempo più resistenti al rumore. Carrot sembra gestire meglio di Google l'effetto di disturbo del sistema e l'accrescere del numero dei termini nell'interrogazione. Si notino i valori nella variante parola chiave e sole parole rilevanti (K + RelWs), che superano addirittura e di molto quelli di Google. Questo induce dunque a pensare che l'approccio *web clustering* è più resistente allo stimolo rumoroso o comunque meglio lo gestisce. Medesime asserzioni possono compiersi, valutando i valori dove si ha un maggior grado di disturbo con un termine casuale inserito nella query (K + Rand + Def).

Hakia risulta, al contrario, altamente perturbabile dalla rumorosità dello stimolo, ma al contempo resistente alla povertà espressiva di questo.

Scendendo nel dettaglio dei dati, seguono una serie di tabelle con i riscontri effettuati. Nelle prime tabelle è possibile valutare per ogni singola variante il grado di precisione, e dunque il comportamento del motore.

Nella prima valutazione non può che notarsi come la *query*, esplicita con il solo termine-chiave, permetta il perseguimento di valori percentuali molto alti. In particolare si coglie la qualità nettamente superiore di Google rispetto agli altri due motori.

Colpisce come il valore più alto per il motore si abbia con l'immissione della sola parola chiave, che però non sembra avere un ruolo così determinante quando si unisce alla definizione o alle sole parole rilevanti. Non permette una crescita in precisione dei risultati che crolla notevolmente con l'inserimento della parola di disturbo, mediamente valida se si considera il solo inserimento di parole rilevanti, causa dunque di una complessità maggiore nel restringere l'esatto dominio indagato.

A dimostrazione di ciò è l'elevato valore che si ha nelle *query* per la variante *DefinitionOnly*, che mostra come un motore non dichiaratamente semantico compia uno grande sforzo in tal senso.

Il contrario avviene invece per il motore di ricerca, semantico e aziendale, Hakia che delude completamente ogni aspettativa, non of-

frendo che risultati mediocri per la sola query diretta costituita dal termine chiave. In questo unico caso considerabile, il risultato di Hakia è inferiore quasi del 50% rispetto a Google, mentre si rivela per il più alto grado di rilevanza dei risultati con la *query* costituita dal solo termine chiave migliore di Carrot.

Carrot si dimostra però un motore particolarmente robusto, pur non raggiungendo gli elevati valori di Google in precisione. Affronta le *query* prive della parola chiave e con le sole parole selezionate per rilevanza con un maggior grado di precisione, nonché meglio fronteggia l'inserimento di parole di disturbo nella *query*.

Appendice

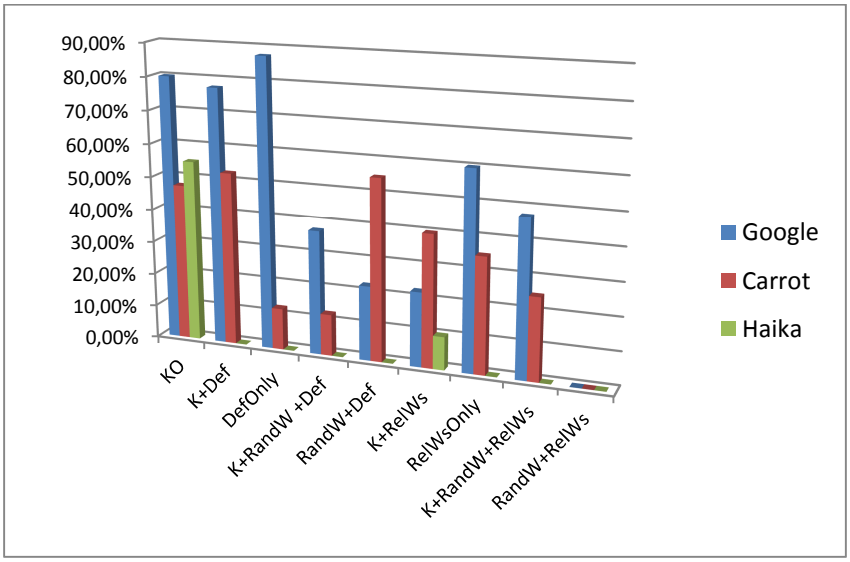
A1. Legenda delle varianti di *query* utilizzate.

Query Type	Description
KO	Keyword Only
K+Def	Keyword and Full text of the Definition
DefOnly	Only the Full text of the Definition
K+RandW+Def	Keyword, Full Text of the Definition and a Randomly selected word
RandW+Def	A Randomly selected word and the Full Text of the Definition
K+RelWs	Keyword and Relevant Words of the definition
RelWsOnly	Only the Relevant Words of the definition
K+RandW+RelWs	Keyword, Full Text of the Definition and Relevant Words of the definition
RandW+RelWs	A Randomly selected word and Relevant Words of the definition

A2. Tabella di rilevazione della percentuale di precisione per singola *query* grado con rilevanza sopra 1 (mediamente e altamente rilevante).

Google	KO	K+Def	DefOnly	K+RandW +Def	RandW+ Def	K+RelWs	RelW- sOnly	K+Rand W+RelWs	RandW+ RelWs
Q1	100,0%	100,0%	100,0%	80,0%	0,0%	0,0%	100,0%	100,0%	0,0%
Q2	100,0%	100,0%	80,0%	30,0%	20,0%	20,0%	50,0%	0,0%	0,0%
Q3	80,0%	70,0%	80,0%	40,0%	30,0%	30,0%	30,0%	50,0%	0,0%
Q4	40,0%	40,0%	90,0%	0,0%	40,0%	40,0%	60,0%	40,0%	0,0%
TOT	**80,0%**	**77,5%**	**87,5%**	**37,5%**	**22,5%**	**22,5%**	**60,0%**	**47,5%**	**0,0%**
Carrot	**KO**	**K+Def**	**DefOnly**	**K+RandW +Def**	**RandW+ Def**	**K+RelWs**	**RelW- sOnly**	**K+Rand W+RelWs**	**RandW+ RelWs**
Q1	50,0%	90,0%	20,0%	20,0%	100,0%	100,0%	40,0%	70,0%	0,0%
Q2	90,0%	90,0%	0,0%	0,0%	60,0%	10,0%	50,0%	0,0%	0,0%
Q3	30,0%	30,0%	30,0%	30,0%	40,0%	30,0%	40,0%	30,0%	0,0%
Q4	20,0%	0,0%	0,0%	0,0%	20,0%	20,0%	10,0%	0,0%	0,0%
TOT	**47,5%**	**52,5%**	**12,5%**	**12,5%**	**55,0%**	**40,0%**	**35,0%**	**25,0%**	**0,0%**
Haika	**KO**	**K+Def**	**DefOnly**	**K+RandW +Def**	**RandW+ Def**	**K+RelWs**	**RelW- sOnly**	**K+Rand W+RelWs**	**RandW+ RelWs**
Q1	30,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q2	70,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q3	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q4	50,0%	0,0%	0,0%	0,0%	0,0%	40,0%	0,0%	0,0%	0,0%
TOT	**55,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**10,0%**	**0,0%**	**0,0%**	**0,0%**

A3. Grafico di comparazione del grado di precisione rispetto ai soli documenti ritenuti mediamente e altamente rilevanti.



A4. Tabella di rilevazione della percentuale di precisione per singola *query* grado con rilevanza sopra 2 (altamente rilevante)

Google	KO	K+Def	DefOnly	K+RandW +Def	RandW+ Def	K+RelWs	RelW- sOnly	K+Rand W+RelWs	RandW+ RelWs
Q1	80,0%	80,0%	100,0%	20,0%	0,0%	0,0%	50,0%	90,0%	0,0%
Q2	90,0%	90,0%	60,0%	30,0%	0,0%	0,0%	40,0%	0,0%	0,0%
Q3	50,0%	0,0%	40,0%	10,0%	30,0%	30,0%	30,0%	30,0%	0,0%
Q4	0,0%	0,0%	10,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
TOT	**55,0%**	**42,5%**	**52,5%**	**15,0%**	**7,5%**	**7,5%**	**30,0%**	**30,0%**	**0,0%**
Carrot	**KO**	**K+Def**	**DefOnly**	**K+RandW +Def**	**RandW+ Def**	**K+RelWs**	**RelW- sOnly**	**K+Rand W+RelWs**	**RandW+ RelWs**
Q1	10,0%	40,0%	0,0%	0,0%	90,0%	50,0%	30,0%	40,0%	0,0%
Q2	60,0%	50,0%	0,0%	0,0%	60,0%	10,0%	50,0%	0,0%	0,0%
Q3	30,0%	30,0%	20,0%	20,0%	40,0%	30,0%	30,0%	0,0%	0,0%
Q4	0,0%	0,0%	0,0%	0,0%	20,0%	10,0%	0,0%	0,0%	0,0%
TOT	**25,0%**	**30,0%**	**5,0%**	**5,0%**	**52,5%**	**25,0%**	**27,5%**	**10,0%**	**0,0%**
Haika	**KO**	**K+Def**	**DefOnly**	**K+RandW +Def**	**RandW+ Def**	**K+RelWs**	**RelW- sOnly**	**K+Rand W+RelWs**	**RandW+ RelWs**
Q1	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q2	60,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q3	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Q4	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
TOT	**40,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**

A5. Grafico di comparazione del grado di precisione rispetto i soli documenti ritenuti altamente rilevanti