Argo CCMS versione 2.7.0.0
Funzionamento della Ricerca Full Text

La ricerca full text in SQL Server si basa sull'algoritmo di ranking denominato BM25 (Best Match 25), algoritmo utilizzato da tutti i motori di ricerca Full Text. Rispetto ad una ricerca di una sequenza esatta di caratteri in un testo, la ricerca Full Text lavora attraverso un criterio di somiglianza.

La somiglianza non consiste nella somiglianza di significato o semantica del testo, ma nella presenza di parole chiave.

La ricerca per somiglianza nella Full Text Search prevede dei passi di elaborazione della frase da cercare, per estrarre le parole chiave, prima di effettuare la ricerca vera e propria.

Supponiamo che la frase da cercare sia:

"`Nelle macchine utensili, non catalogate`".

Ecco i passaggi che l'algoritmo di ricerca effettua ancora prima di iniziare la ricerca nella base di dati:

- Esclusione della punteggiatura:
  `Nelle macchine utensili non catalogate`
  È stata esclusa la virgola. Questo indica che la punteggiatura non ha nessuna importanza nel testo cercato, ne nel testo da cercare presente nella base di dati.

- Esclusione delle congiunzioni degli articoli e di tutte le parole di collegamento:
  `macchine utensili catalogate`
  "Nelle" e "non" sono state escluse. In questo caso si nota che l'aspetto di essere una frase in negativo ("non") viene perso. Saranno escluse anche alcune parole troppo frequenti nella lingua italiana, come la parola "è".

- Di ogni parola rimasta viene estratta la radice, secondo le regole della lingua italiana:
  `macchin / utensil / catalog`
  L'estrazione della radice permetterà di estendere la ricerca per includere le declinazioni di quella parola, per esempio macchina o macchine.
  Oppure le coniugazioni di un verbo: catalogare, catalogato. Inoltre la parola chiave "catalog" sarà cercata sia come sostantivo che come verbo.
  Da questo punto in poi la ricerca considera le varie parole chiave come entità a se stanti.

- La lista di parole chiave viene riordinata in ordine alfabetico e vengono eliminati i duplicati (in questo caso non erano presenti duplicati):
  `catalog / macchin / utensil`

Ora che dalla frase inserita sono state estratte le parole chiave, la ricerca procede con la scansione del testo nei record del database: ad ogni record viene assegnato un punteggio (Rank) che determinerà la posizione del record nei risultati della ricerca. Record con alto punteggio appariranno in cima alla lista dei risultati, record con punteggio troppo basso saranno scartati.

Ad ogni ricerca il punteggio viene riscalato sulla base del punteggio minimo e massimo raggiunto in quella specifica ricerca.

L'algoritmo di ricerca BM 25 è definito rigorosamente a livello matematico, riportiamo la formula solo per indicare che l'algoritmo è deterministico: dato un database D e una frase Q da cercare il risultato sarà sempre uguale. Variando anche minimamente i dati nel database o la frase cercata i risultati possono essere totalmente differenti.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Figura 1: Tutto chiaro, vero?

Proviamo quindi a dare un significato comprensibile alla formula dell'algoritmo.

L'algoritmo di attribuzione del ranking BM25 può essere definito in maniera intuitiva secondo questi punti:

1. **Rilevanza delle parole chiave**: SQL Server cerca di valutare quanto un documento è pertinente in base alla corrispondenza delle parole chiave. Anche se non tutte le parole cercate sono presenti, il motore può assegnare un ranking alto se trova corrispondenze parziali ma rilevanti (parole più rare hanno un punteggio maggiore).
2. **Lunghezza del documento**: BM25 penalizza i documenti molto lunghi: un testo più lungo potrebbe avere più opportunità di contenere le parole cercate ma senza essere necessariamente più rilevante. Un testo più breve, invece, potrebbe risultare più conciso e concentrato sulle parole chiave, ricevendo un punteggio più alto.
3. **Distribuzione delle parole chiave**: non tutte le parole chiave hanno lo stesso peso. Alcune potrebbero essere considerate più "importanti" in base alla loro frequenza relativa all'interno del singolo record e in tutti i record indicizzati nel database.
4. **Frequenza delle parole chiave nel documento**: record dove le parole chiave appaiono più frequentemente tendono ad avere un ranking migliore, purché la frequenza non sia così alta da far sembrare il record meno significativo.
5. **Documenti più concisi**: un testo breve ma preciso può ottenere un punteggio più alto rispetto a un testo lungo dove le parole chiave sono più sparse. La precisione e la densità delle parole chiave nel testo giocano un ruolo cruciale.

Riassumendo anche se un record non contiene tutte le parole chiave, ottiene un ranking migliore perché:

- È più conciso attorno alle parole chiave (punto 2, punto 5)
- Il conteggio totale delle parole chiave nel record è molto alto (punto 3), magari è una stessa, o poche, parole chiave ripetute molte volte nello stesso record (punto 4)
- Le parole chiave trovate nel record sono state considerate più importanti rispetto all'intero database (punto 1, punto 3).

All'opposto un record può ottenere un punteggio basso perché:

- È molto lungo
- Non contiene in generale in un numero significativo di parole chiave
- Contiene parole chiave che hanno un punteggio basso nel database (troppo diffuse a livello di database)
- Contiene una ripetizione eccessiva di una o poche parole chiave, anche se hanno un ottimo punteggio. Queste parole chiave, anche se hanno un punteggio alto a livello di database (sono poco diffuse nel database), vengono declassate SOLO per questo specifico record perché troppo ripetute all'interno del record.

Vediamo con degli esempi pratici l'effetto di questi principi cercando la stessa frase.

Esempio 1

record A:	record B:
"**Macchin**e agricole"	"E' necessario prestare una maggiore attenzione alla manutenzione delle **macchine utensili**"
PUNTEGGIO SUPERIORE	

Il record A è più conciso e è costituito al 50 % di parole chiave, mentre nel record B invece, che ha ben due parole chiave, la presenza di parole chiave è inferiore al 50% del testo.

Esempio 2

record A:	record B:
"**Macchine** agricole, **Macchine** Industriali, **Macchine** Automatiche"	"E' necessario prestare una maggiore attenzione alla manutenzione delle **macchine utensili**"
PUNTEGGIO SUPERIORE	

Il record A contiene 3 istanze di parola chiave (la stessa) rispetto alle 2 istanze di parole chiave del record B.

Esempio 3 (identico all'esempio 2, in un database in cui il termine "Macchine" è molto presente)

record A:	record B:
"**Macchine** agricole, **Macchine** Industriali, **Macchine** Automatiche"	"E' necessario prestare una maggiore attenzione alla manutenzione delle **macchine utensili**"
	PUNTEGGIO SUPERIORE

In questo terzo esempio è successo che il database nella sua interezza contiene parecchi record con la parola chiave "Macchine". In questo caso la parola chiave "Macchine" viene declassata e la ricerca ha premiato il record B per la sola presenza della parola chiave "Utensili" che, essendo meno presente nell'intero database, ha un valore più alto.
Questo terzo esempio ci fa notare come la ricerca Full Text possa dare risultati diversi in seguito all'evoluzione dei dati nel database (da un certo momento in poi inserisco molti record con la parola "Macchine", che viene declassata a parola poco importante perché troppo diffusa).