8. Ranking

Limiti del modello booleano

Finora, le nostre query sono state basate sul **modello booleano**, in cui i documenti o corrispondono perfettamente alla query o non corrispondono affatto. Questo approccio è utile in alcuni contesti:

- È adatto per **utenti esperti**, che hanno una chiara comprensione delle loro esigenze e della struttura della collezione di documenti.
- È utile per le **applicazioni automatizzate**, che possono elaborare migliaia di risultati per ulteriori analisi.

Tuttavia, il modello booleano presenta diversi problemi per la maggior parte degli utenti:

- **È difficile da usare**: molti utenti non sanno scrivere query booleane o le trovano troppo complesse e macchinose.
- **Genera risultati sbilanciati**: spesso restituisce **troppi** risultati (rendendo difficile trovare le informazioni rilevanti) o **troppo pochi** (fino a zero risultati).
- **Non è ideale per la ricerca sul web**, dove gli utenti si aspettano risposte pertinenti ordinate per rilevanza, anziché un semplice insieme di documenti che soddisfano esattamente la query.

Esempio. Supponiamo che un utente stia cercando una soluzione per un problema con la sua scheda di rete **D-Link 650**.

- **Query 1:** `"standard user dlink 650"` → **200.000 risultati**, troppi per essere utili.
- **Query 2:** `"standard user dlink 650 no card found"` → **0 risultati**, troppo restrittiva.

Trovare un equilibrio tra una query che **non restituisca troppi documenti** e una che **non sia troppo restrittiva** richiede **abilità e tempo**, rendendo il modello booleano poco pratico per la maggior parte degli utenti.

- L’operatore **AND** può restringere troppo la ricerca, portando a pochi o nessun risultato.
- L’operatore **OR** può ampliare eccessivamente la ricerca, generando un numero eccessivo di risultati.

A causa di questi limiti, i sistemi moderni di recupero dell'informazione adottano **modelli più sofisticati**, come quelli

Recupero basato su ranking

A differenza del modello booleano, in cui il sistema restituisce un insieme di documenti che soddisfano esattamente la query, il **recupero basato su ranking** (**ranked retrieval**) organizza i documenti in un **ordinamento decrescente di rilevanza** rispetto alla query:

- Non è necessario restituire l'intero insieme di risultati: il sistema mostra solo i **top** k **risultati** (solitamente $k \approx 10$).
- L'utente non viene sopraffatto da un numero eccessivo di documenti irrilevanti.
- Il problema dei **grandi insiemi di risultati** non è più un ostacolo, poiché solo i documenti più rilevanti vengono visualizzati.

Il recupero basato su ranking è efficace **solo se il modello di ranking funziona bene**, ossia se riesce a ordinare i documenti in modo che i più pertinenti appaiano ai primi posti.

Uno dei modelli di ranking più noti e utilizzati è il **modello dello spazio vettoriale** (**Vector Space Model**), che misura la somiglianza tra la query e i documenti attraverso rappresentazioni vettoriali e misure come la **similarità coseno**.

L'adozione del ranking ha reso la ricerca più efficiente e user-friendly, migliorando notevolmente l'esperienza dell'utente rispetto al tradizionale modello booleano.

Modello dello spazio vettoriale (Vector Space Model - VSM)

Il **modello dello spazio vettoriale** (**VSM**) rappresenta i documenti in modo simile al modello **bag-of-words**, utilizzando un **vettore di pesi dei termini**.

Struttura del modello

- Ogni **documento** è rappresentato come un vettore, in cui ogni componente corrisponde a un **peso associato a un termine**.
- L'intera **collezione di documenti** può essere rappresentata come una **matrice** di pesi dei termini.

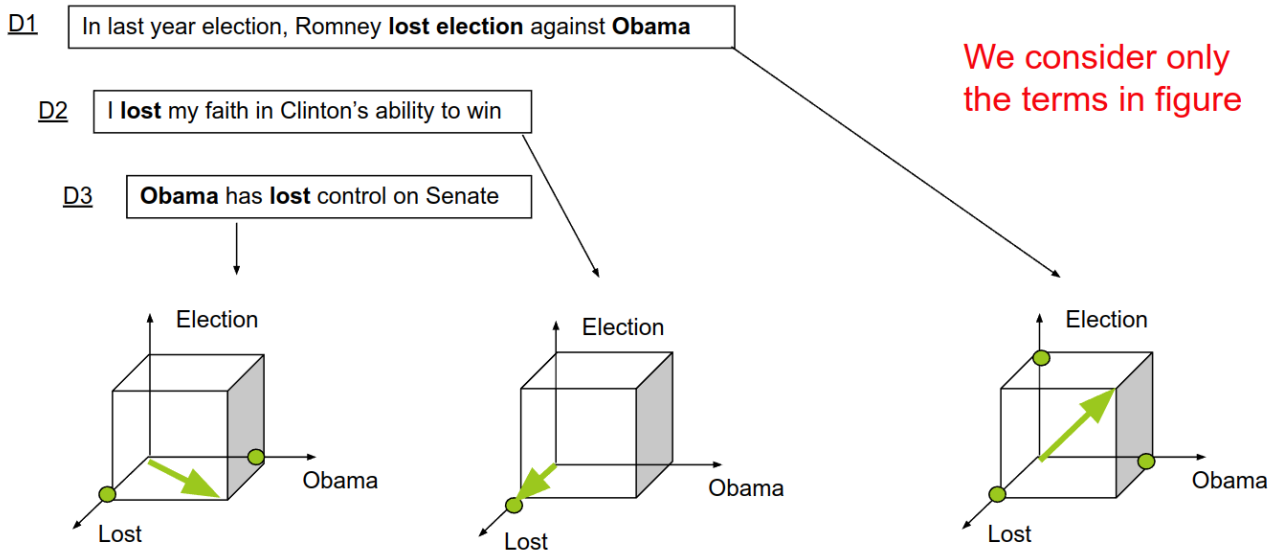
Se indichiamo con d_{ij} il peso del **termine** j nel **documento** i, la collezione può essere formalmente espressa come una matrice in cui:

$$d_{ij} = \text{peso del termine } j \text{ nel documento } i$$

Questa rappresentazione permette di calcolare la **similarità tra query e documenti** attraverso misure come la **similarità coseno**, consentendo di ordinare i risultati in base alla loro rilevanza.

Il VSM è una delle tecniche più utilizzate nel recupero delle informazioni grazie alla sua capacità di gestire grandi collezioni di documenti in modo efficace.

Esempio. 3 termini (spazio vettoriale tridimensionale) e 3 documenti.



Documenti come vettori

Abbiamo quindi uno spazio vettoriale di dimensione $|V|$. I termini rappresentano gli assi dello spazio, mentre i documenti corrispondono a punti o vettori all'interno di questo spazio. Questo spazio è caratterizzato da una dimensionalità estremamente elevata: nell'applicazione ai motori di ricerca web, si possono avere decine di milioni di dimensioni. I vettori risultano essere molto sparsi, poiché la maggior parte delle loro componenti è pari a zero.

Interrogazioni come vettori

Un'idea chiave consiste nel rappresentare anche le interrogazioni come vettori nello stesso spazio. Successivamente, i documenti vengono classificati in base alla loro prossimità alla query in questo spazio. La prossimità tra documenti e query viene misurata attraverso la similarità tra vettori e può essere approssimata come l'inverso della distanza.

Questa tecnica consente di superare il modello booleano, che classifica i documenti in modo rigido come rilevanti o non rilevanti. Invece, si adotta un approccio in cui i documenti più pertinenti vengono classificati con un punteggio più alto rispetto a quelli meno rilevanti.

Riepilogo

Nel modello booleano, un documento è considerato esclusivamente rilevante o non rilevante. Tuttavia, rappresentando documenti e interrogazioni come vettori, è possibile calcolare la similarità tra il vettore di un documento e quello di una query. Questo valore rappresenta una stima della rilevanza del documento rispetto alla query.

Per fare ciò, è necessario definire una misura di similarità tra vettori, ma prima di procedere è fondamentale introdurre il concetto di frequenza dei termini.

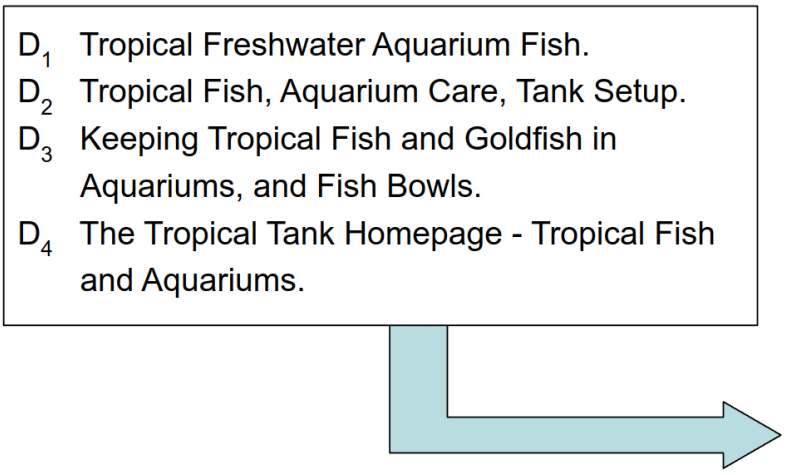
TF: Frequenza dei termini

Fino a questo punto, i vettori erano binari, rappresentando semplicemente la presenza o l'assenza di un termine in un documento. Tuttavia, vorremmo assegnare un peso a ciascun termine nello spazio vettoriale.

La frequenza di un termine, indicata come $tf_{t,d}$, è definita come il numero di volte in cui il termine t compare nel documento d. Questo valore viene utilizzato per calcolare il punteggio di corrispondenza tra una query e un documento.

Esempio. TF come peso del termine

Le rappresentazioni tridimensionali sono utili per visualizzare i dati, ma possono risultare fuorvianti quando si considera uno spazio ad alta dimensionalità, ossia con più di tre termini.



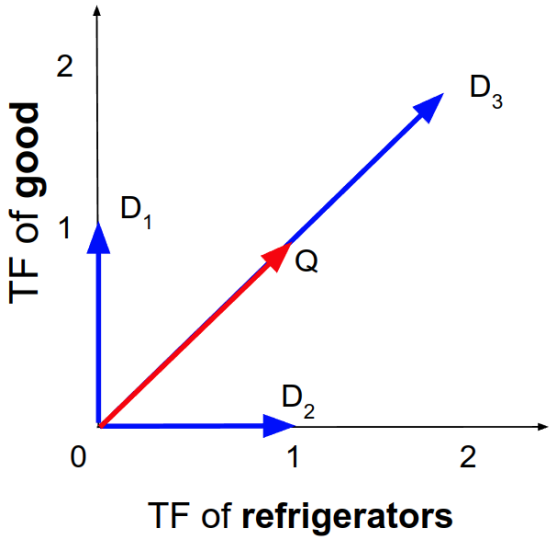
	Documents			
Terms	**D_1**	**D_2**	**D_3**	**D_4**
Aquarium	1	1	1	1
Bowl	0	0	1	0
Care	0	1	0	0
Fish	1	1	2	1
Freshwater	1	0	0	0
Goldfish	0	0	1	0
Homepage	0	0	0	1
Keep	0	0	1	0
Setup	0	1	0	0
Tank	0	1	0	1
Tropical	1	1	1	2

Distanza nello spazio vettoriale

Come possiamo formalizzare la prossimità tra due vettori quando utilizziamo le frequenze dei termini?

Esempio. Consideriamo la query Q: **"Good refrigerators"** e i seguenti documenti:

- D_1: *"Good morning to all of you."*
- D_2: *"Don't put pizza in refrigerators."*
- D_3: *"Good Refrigerator Review: top five good refrigerators."*



In questo esempio, stiamo visualizzando solo le due dimensioni più rilevanti: **"good"** e **"refrigerators"**. Tuttavia, esiste una dimensione per ogni termine indicizzato, come **"pizza"**, **"morning"**, **"review"**, ecc.

L'obiettivo è misurare la similarità tra la query e i documenti, utilizzando le frequenze dei termini per calcolare una distanza nello spazio vettoriale.

Prima approssimazione: Distanza euclidea

Una possibile misura di prossimità tra vettori è la distanza in linea retta tra i loro estremi, ovvero la **distanza euclidea**.

Se applichiamo questa metrica all'esempio precedente, otteniamo i seguenti risultati:

- La distanza tra Q e D_1 è **1**
- La distanza tra Q e D_2 è **1**
- La distanza tra Q e D_3 è $\sqrt{2} \approx 1.414$

Tuttavia, secondo questa metrica, il documento D_3 risulta essere il meno simile alla query, il che sembra errato! Ricordiamo che la query è **"Good refrigerators"** e il documento D_3 contiene ripetutamente entrambi i termini, mentre D_1 e D_2 no.

Questo suggerisce che la distanza euclidea potrebbe non essere la misura ideale per valutare la similarità tra documenti e query nello spazio vettoriale.

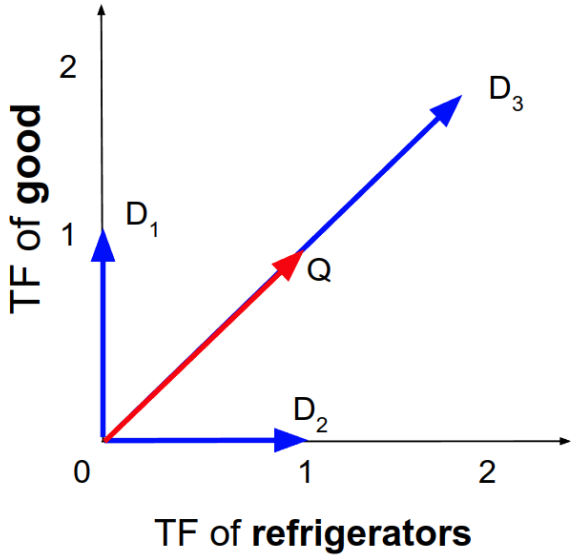
Angolo invece di distanza

Immaginiamo un esperimento mentale: prendiamo un documento d e lo ripetiamo molte volte concatenandolo a se stesso. Chiamiamo questo nuovo documento più lungo d'. Dal punto di vista semantico, i contenuti di d e d' sono identici.

Tuttavia, se misuriamo la distanza euclidea tra d e d', questa può risultare molto grande, nonostante il fatto che il contenuto non sia realmente cambiato. D'altra parte, se consideriamo l'**angolo tra i due vettori**, esso rimane **0**, corrispondendo a una similarità massima.

Questa osservazione porta a un'idea chiave: invece di classificare i documenti in base alla distanza euclidea rispetto alla query, è più efficace ordinarli in base all'**angolo** tra il loro vettore e quello della query. Un angolo più piccolo indica una maggiore similarità tra documento e query.

Esempio. Ora consideriamo l'**angolo tra i vettori** per misurare la similarità tra la query e i documenti.



Applicando questa metrica all'esempio precedente, otteniamo:

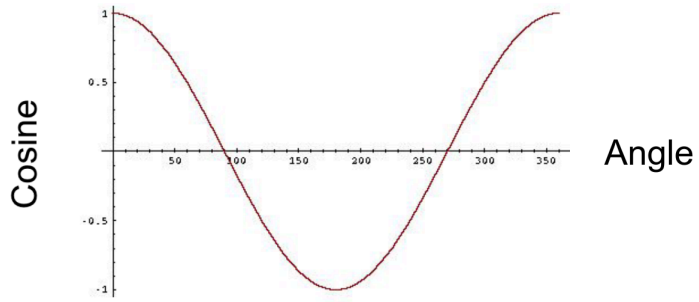
- L'angolo tra Q e D_1 è **45°**
- L'angolo tra Q e D_2 è **45°**
- L'angolo tra Q e D_3 è **0°**

In questo caso, il documento D_3 risulta il **più simile** alla query, ed è esattamente il comportamento desiderato. Questo metodo riflette meglio la somiglianza semantica tra i testi rispetto alla distanza euclidea.

L'approccio basato sugli angoli ci permette di classificare correttamente i documenti, assegnando una maggiore rilevanza a quelli che condividono più termini significativi con la query.

Dagli angoli ai coseni

Le seguenti due nozioni sono equivalenti: ordinare i documenti in ordine crescente dell'angolo tra la query e il documento, oppure ordinare i documenti in ordine decrescente del coseno tra la query e il documento. Il coseno è una funzione monotonicamente decrescente nell'intervallo da $0°$ a $180°$, il che significa che è inversamente proporzionale all'ampiezza dell'angolo.



Similarità del coseno

La similarità del coseno è definita come segue

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}|\left|\vec{d}\right|} = \frac{\vec{q}}{|\vec{q}|}\cdot\frac{\vec{d}}{\left|\vec{d}\right|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

Dot product

Unit vectors

Dove q_i rappresenta il peso *tf* del termine i nella query, e d_i il peso *tf* del termine i nel documento. Tale valore rappresenta il coseno dell'angolo tra i vettori $\vec{q}$ e $\vec{d}$.

Rarità dei termini

I termini comuni sono meno informativi rispetto a quelli rari. Si consideri, ad esempio, una query come "good refrigerators". Con la frequenza del termine (TF, term frequency), si analizza la frequenza delle parole all'interno di ciascun documento. Tuttavia, la parola "good" è molto comune e si trova nella maggior parte dei documenti, mentre "refrigerators" è meno comune e quindi più informativa per determinare la rilevanza. È quindi opportuno attribuire un peso maggiore alle parole rare rispetto a quelle comuni. Per catturare questo concetto, si utilizza l'**inverse document frequency** (IDF), che si basa sulla **document frequency** (DF).

DF: Frequenza nei documenti

La **document frequency** di un termine t, indicata come df_t, rappresenta il numero di documenti che contengono il termine t. Nonostante il nome, la document frequency riguarda l'intera collezione di documenti. Essa misura quanto un termine è comune all'interno della collezione. Il valore df_t è una misura inversa dell'informatività del termine t: quanto più comune è una parola, tanto meno sarà utile per determinare la rilevanza (caso estremo: le

stop words). Si utilizza quindi il valore df per formulare il concetto inverso e rappresentare la rarità di una parola.

IDF: Frequenza inversa nei documenti

La **inverse document frequency** (IDF) di un termine t è definita come:

$$idf_t = \log_{10}\left(\frac{N}{df_t}\right)$$

dove N è il numero totale di documenti nella collezione. Si utilizza il logaritmo $\log(N/df_t)$ al posto di N/df_t per attenuare l'effetto dell'IDF. Ad esempio, una parola comune come "home" può avere una frequenza anche un milione di volte maggiore rispetto a una parola rara come "fenestration". L'uso del logaritmo serve ad appiattire questo fenomeno esponenziale della linguistica (descritto dalla legge di Zipf), evitando che parole molto comuni abbiano un IDF pari a zero.

Effetto dell'IDF sul ranking

L'IDF ha un effetto sul ranking solo per le query composte da almeno due termini. Per una query composta da un solo termine, come ad esempio "iPhone", l'IDF non influisce sull'ordinamento dei risultati. Tuttavia, per query come "capricious person", il peso dato dall'IDF fa sì che le occorrenze della parola "capricious" abbiano molto più valore nel ranking finale rispetto alle occorrenze di "person". L'IDF non viene mai utilizzato da solo, ma sempre in combinazione con la TF per stimare meglio la rilevanza di un documento.

TF-IDF

Il peso **tf-idf** di un termine è il prodotto tra il suo peso tf e il suo peso idf, ed è definito dalla formula:

$$w_{t,d} = tf_{t,d} \times \log_{10}\left(\frac{N}{df_t}\right)$$

Si tratta dello schema di pesatura più noto nel campo del recupero dell'informazione. Nota bene: il trattino in "tf-idf" è un **trattino tipografico** (hyphen), non un segno meno. Nella letteratura si possono trovare anche denominazioni alternative come **tf.idf** o **tf × idf**.

Il peso tf-idf di un termine aumenta sia con il numero di occorrenze del termine all'interno di un documento (cioè con la tf), sia con la rarità del termine all'interno della collezione (cioè con l'idf).

Modelli di recupero

Esistono diversi modelli classici di recupero dell'informazione, tra cui il **modello booleano**, il **modello dello spazio vettoriale** e i **modelli probabilistici**. Tra questi ultimi, uno dei più noti è il **BM25**. Esistono inoltre modelli basati sul linguaggio (**language models**). I progressi

nei modelli di recupero sono stati spesso associati a miglioramenti in termini di efficacia del recupero stesso.

Modelli probabilistici

Il principio della classificazione probabilistica nel recupero dell'informazione è stato formulato da Robertson nel 1977. Secondo tale principio, l'ordinamento dei documenti dovrebbe basarsi sulla **probabilità di rilevanza**, stimata nel modo più accurato possibile a partire dai dati disponibili. Questo approccio garantisce la **massima efficacia ottenibile** con le informazioni a disposizione. In altre parole:

"[...] Se l'ordinamento dei documenti è effettuato in base alla probabilità decrescente di rilevanza (dove le probabilità sono stimate nel modo più accurato possibile a partire dai dati disponibili), l'efficacia complessiva sarà la migliore ottenibile sulla base di quei dati."

Okapi BM25

Lo schema di pesatura **BM25**, spesso chiamato **Okapi weighting** (dal sistema in cui è stato originariamente implementato), è stato sviluppato come un modello probabilistico che tiene conto di tre fattori fondamentali:

- la **frequenza del termine** (term frequency) all'interno del documento
- la **rarità del termine** nel corpus, in modo simile all'IDF
- la **lunghezza del documento**

Il BM25 è diventato un algoritmo di ranking molto popolare ed efficace, pur mantenendo solide basi probabilistiche. Le formule di pesatura dei termini secondo BM25 sono state utilizzate con successo in un'ampia varietà di collezioni e compiti di ricerca.

Okapi BM25: saturazione della TF

Nel modello TF-IDF, un documento può ricevere un punteggio di similarità molto elevato se la frequenza del termine della query è molto alta. Tuttavia, nel modello **Okapi BM25**, il punteggio di similarità è sempre compreso tra **0 e 1**. A un certo punto, l'aumento della frequenza del termine non contribuisce più in modo significativo al punteggio, grazie al meccanismo di **saturazione della TF**, che limita l'effetto delle ripetizioni eccessive del termine nel documento.

Okapi BM25: parametri

Il modello Okapi BM25 prevede due parametri fondamentali:

Il parametro k_1 controlla la rapidità con cui un aumento nella frequenza del termine porta alla saturazione della frequenza stessa. In pratica, determina la **pendenza della curva** tf() del BM25. Il valore predefinito di k_1 è **1.2**. Valori più bassi determinano una saturazione più rapida, mentre valori più alti comportano una saturazione più lenta.

Il parametro b regola l'effetto della **normalizzazione rispetto alla lunghezza del documento**. Il valore predefinito è **0.75**. Un valore pari a **0.0** disattiva completamente la normalizzazione, mentre un valore pari a **1.0** applica una normalizzazione completa in base alla lunghezza.

Modello linguistico

Un **modello linguistico statistico** assegna una probabilità a una sequenza di m parole $P(w_1, \ldots, w_m)$ attraverso una distribuzione di probabilità. A ciascun documento di una collezione viene associato un modello linguistico separato. I documenti vengono classificati in base alla **probabilità che la query Q sia generata dal modello linguistico del documento** M_d, ossia in base a $P(Q \mid M_d)$.

Nel **modello linguistico unigramma**, la distribuzione di probabilità è definita sui singoli termini del linguaggio. La generazione del testo consiste nell'estrarre parole da un "contenitore" (bucket) secondo la distribuzione di probabilità, con reinserimento dopo ogni estrazione.

Esistono anche **modelli linguistici N-gramma**, come quelli bigramma o trigramma, in cui la probabilità di una parola dipende dalle parole precedenti. Alcune applicazioni sfruttano questi modelli per una maggiore precisione contestuale.

Un documento è considerato un buon risultato per una query se il **modello del documento ha un'alta probabilità di generare la query**, il che accade tipicamente se il documento contiene frequentemente le parole della query.

Un modello linguistico può essere visto come un **automa finito** con una misura di probabilità applicata alle stringhe che può generare. La **somma di tutte le probabilità associate ai termini possibili**, inclusa la possibilità di **interrompere** la generazione, è **uguale a 1**.

Modelli di linguaggio basati su probabilità semplici

Il modo più semplice per calcolare la probabilità di una parola in un testo di input è contare quante volte appare e dividere quel numero per il totale delle parole presenti nell'input. Questo tipo di modello linguistico è chiamato **modello Unigramma**, e la probabilità di una parola dipende unicamente dalla sua frequenza nel testo.

Modelli linguistici unigramma

Per trovare la probabilità di generare una frase S composta da n parole, si moltiplica la probabilità che venga generata ciascuna parola, secondo la formula:

$$P(S) = P(w_0)P(w_1)\ldots P(w_n)$$

Una limitazione importante di questo modello è che qualsiasi combinazione delle stesse parole avrà la stessa probabilità di comparsa (ad esempio, "two times" e "times two" avranno

la stessa probabilità). Questo modello è il più semplice da calcolare e utilizzare, ma è evidente che si può migliorare questa idea di linguaggio generato “indipendentemente”.

Modello Bigramma

Nel modello bigramma, si considera la probabilità di una parola in base alla parola che la precede. Ad esempio, nella frase “Espresso is awesome, and user-friendly”, i bigrammi sono: “espresso is”, “is awesome”, “and user”, “user friendly”.

Le frasi dei documenti vengono suddivise per creare un dizionario di bigrammi e unigrammi. Si procede poi al conteggio per calcolare la probabilità di ciascun bigramma, il che permette di determinare la probabilità della parola successiva.

Nel caso degli **N-grammi**, un valore grande di n rende il calcolo molto intenso dal punto di vista computazionale. Al contrario, un valore troppo piccolo non permette di catturare dipendenze a lungo termine. I bigrammi sono gli n-grammi più semplici da visualizzare perché la probabilità di co-occorrenza tra due parole può essere rappresentata in una tabella, dove tutte le parole dell’input compaiono sia come righe che come colonne.

Come esempio, si può considerare il **Corpus di frasi del Berkeley Restaurant Project**, composto da 9222 frasi. In questo caso ci si concentra sul calcolo della probabilità dei bigrammi tra 8 parole. Tutte le probabilità sono calcolate utilizzando la stessa formula adottata per gli altri n-grammi.

Modelli linguistici N-grammi

Nel modello linguistico unigramma, le parole non hanno alcuna correlazione tra loro e sono trattate come eventi separati. Tuttavia, nei documenti reali alcune parole appaiono spesso in successione (come ad esempio “Merry Christmas”).

Per modellare questo fenomeno si utilizzano i **modelli linguistici N-grammi**, che condizionano la probabilità di una parola in base ai precedenti $n-1$ termini (dove w_i è la parola all’indice i):

$$P(w_1 \dots w_n) = \prod_{i=2}^{n+1} p(w_i | w_{i-n+1} \dots w_{i-1})$$

Gli N-grammi sono modelli linguistici che correlano n parole tra loro. I modelli con $n \geq 2$ sono usati principalmente in attività come il riconoscimento vocale, la correzione ortografica e la traduzione automatica, dove è necessario calcolare la probabilità di un termine condizionata al contesto circostante, e non sono generalmente utilizzati nei sistemi di recupero dell’informazione (IR).

Ordinamento usando modelli linguistici

Per ciascun documento viene creato un modello in cui si calcola la probabilità di comparsa di ciascuna parola, utilizzando un qualsiasi n-gramma rappresentato con u nella seguente formula:

$$p(w|u) = \frac{c(uw)}{\sum_{w'} c(uw')}$$

dove u varia su tutti i $(n-1)-$grammi.

Quando viene emessa una query, ogni modello calcola la probabilità che la query sia apparsa come un campione casuale del proprio documento. I risultati vengono quindi ordinati in ordine decrescente di probabilità.

Efficienza dei modelli linguistici nel ranking

Il calcolo del punteggio di rilevanza utilizzando modelli non booleani può risultare piuttosto intensivo dal punto di vista computazionale. Un documento può essere recuperato e ricevere un punteggio di rilevanza anche se non contiene tutti i termini della query. Inoltre, l'uso di formule complesse per la rilevanza richiede calcoli aggiuntivi per determinare il punteggio.

Esistono diverse strategie per controllare l'efficienza del processo. Il modo in cui si accede ai documenti incide significativamente sui tempi di calcolo: si può scegliere se elaborare un documento alla volta o un termine alla volta. Nella maggior parte dei casi, non si è interessati ai documenti meno rilevanti, il che consente ulteriori ottimizzazioni.

Tecniche TAAT vs DAAT

La tecnica **TAAT** (Term At A Time) prevede il calcolo dei punteggi per tutti i documenti contemporaneamente, elaborando un termine della query alla volta. Al contrario, la tecnica **DAAT** (Document At A Time) calcola il punteggio totale per ciascun documento, tenendo conto di tutti i termini della query, prima di passare al documento successivo.

Ciascun approccio ha implicazioni specifiche sul modo in cui l'indice di recupero viene strutturato e memorizzato.

Calcolo efficiente del punteggio

Una grande parte del lavoro svolto dalla CPU per ogni query di ricerca è dedicata esclusivamente al calcolo del punteggio di rilevanza. In generale, esiste un vincolo rigido sulla latenza (ad esempio, 250 ms) entro cui bisogna rispondere alla query, altrimenti l'utente non sarà soddisfatto. L'allocazione delle risorse CPU non consente di calcolare esaustivamente il punteggio di ogni documento per ogni query.

L'idea di base è evitare di calcolare il punteggio dei documenti che sicuramente non rientreranno tra i primi K risultati.

Ordinamento sicuro (safe ranking)

Con il termine **“ordinamento sicuro”** si indicano i metodi che garantiscono che i K documenti restituiti siano effettivamente i K con il punteggio più alto in assoluto. Questa garanzia non è limitata solo a metriche come la similarità coseno.

Ma è accettabile utilizzare un approccio non sicuro? Sì, i risultati ottenuti sarebbero approssimati, ma comunque sufficientemente buoni per soddisfare l’utente. L’obiettivo è ridurre il numero di calcoli pur continuando a fornire risultati di buona qualità.

Ordinamento non sicuro (non-safe ranking)

Un ordinamento non sicuro può essere accettabile, poiché la funzione di ordinamento è solo una **stima della soddisfazione dell’utente**. Anche documenti che si trovano appena fuori dai primi K possono risultare adeguati.

Non-sicuro: eliminazione tramite indice

Una strategia è considerare solo i termini della query con alto idf (ovvero rari), scartando quelli comuni. Un’altra è considerare solo i documenti che contengono molti termini della query, escludendo quelli che ne contengono solo pochi.

Non-sicuro: liste dei campioni (Champion lists)

Per ciascun termine, i punteggi dei documenti più rilevanti vengono pre-calcolati **prima** che venga emessa qualsiasi query. Questo permette di ridurre il carico computazionale al momento della ricerca, concentrandosi solo su un sottoinsieme di documenti potenzialmente rilevanti.