Sistema IR per collezione di articoli medici utilizzando diversi insiemi di stopword

Alessandro Stefani[1], Caterina Buranelli[2], and Cristi Gutu[3]

[1] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1148387
alessandro.stefani.6@studenti.unipd.it
[2] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1163443
caterina.buranelli@studenti.unipd.it
[3] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1147351
gheorghecristi.gutu@studenti.unipd.it

Sommario Questo progetto tratta la realizzazione attraverso il pacchetto *Whoosh*, di un motore di ricerca volto al reperimento di documenti della collezione sperimentale *OHSUMED* indicizzata opportunamente. Si è affrontato il problema della valutazione del sistema di ricerca studiando l'effetto dell'esclusione di diversi insiemi di stopword. Nella relazione verrà evidenziato come l'uso delle stopword migliori l'efficacia del sistema IR. Il progetto è anche corredato di un webserver che permette all'utente di interrogare il motore di ricerca in forma interattiva, attraverso un browser a scelta.

Keywords: *Information Retrieval · IR · statistica · reperimento · indicizzazione · Whoosh · Python · webserver · web · interrogazione web · stopword*

1 Introduzione

Lo scopo finale di un Sistema di Information Retrieval (IRS) è quello di reperire documenti rilevanti relativi a una certa esigenza informativa[4]; dunque i documenti sono il primo input del sistema, mentre il secondo è costituito dalle interrogazioni; i documenti devono essere indicizzati e nell'indice creato si andrà a effettuare la ricerca per reperire documenti rilevanti[5].

Questa seconda parte è detta reperimento e non si occupa solo di ricercare tra i documenti, ma anche di riordinare secondo un certo ordine di rilevanza.

Ciò che descrive i documenti e ciò che descrive le interrogazioni deve essere confrontabile, infatti nei programmi di indicizzazione e di reperimento si usa uno schema, che si deve tenere in considerazione per entrambi i casi.

[4] insieme delle circostanze in cui una persona ha un problema da risolvere o un compito da svolgere richiede informazioni importanti, utili o necessarie per la risoluzione del problema o lo svolgimento del compito

[5] la rilevanza è la proprietà che rende l'informazione importante, utile o necessaria a soddisfare l'esigenza informativa dell'utente

L'indicizzazione è un trade-off tra il miglioramento della rappresentazione del contenuto informativo dei documenti (efficacia) e la gestione degli indici (efficienza), in questa relazione si tratterà il problema dell'efficacia. Esistono diversi modi per risolvere questo problema; nel nostro caso ci concentriamo sull'utilizzo di stopword.

Il resto della relazione è organizzato come segue. Nella sezione 2, si descrivono alcuni dei metodi standard, i più didattici, su cui si basano gli esperimenti. La sezione 3, presenta l'interfaccia web e le sue modalità di utilizzo. Nella sezione 4, sono riportati gli esperimenti fatti ed i risultati ottenuti, commentati e sotto forma di tabelle esplicative. Infine, nella sezione 5, si traggono delle brevi conclusioni.

2 Base di partenza

Un sistema IR si può basare su vari modelli e metodi di rappresentazione delle informazioni e di reperimento. Come già detto, in questa sezione si presenteranno quelli scelti per implementare il motore di ricerca utilizzato negli esperimenti.

2.1 Indicizzazione

L'indice creato da whoosh rappresenta la collezione utilizzando un così detto *inverted index*[5] ovvero, una "matrice" in cui si associa ad ogni termine una lista dei dati rilevanti corrispondenti.

I dati rilevanti possono essere, per esempio, i documenti in cui sono presenti i termini, con le rispettive frequenze. Ogni elemento della lista è detto *posting* e la parte del *posting* che fa riferimento ad uno specifico documento è detta *pointer*.

In qesto progetto si tratta in particolare dell'eliminazione delle *stopword*[2].

Questa è una procedura di elaborazione del testo, se utilizzata si deve fare sia sui documenti da indicizzare che eventualmente sulle query inviate al motore di ricerca.

Le stopword sono parole che non portano informazione significativa da sole e quindi non aiutano a capire se un documento è rilevante o meno. Inoltre, solitamente si tratta di parole molto comuni nella collezione che quindi risultano in un costo rilevante dal punto di vista dello spazio di memoria occupato.

Parole che di solito si identificano come stopword nella lingua inglese sono per esempio: "the", "a", "an", "that" o "those".

Oltre a queste parole, che non hanno significato a se stante, si possono identificare anche stopword a seconda del contesto. Con questo si intende che delle parole possono diventare stopword in certi contesti anche se in altri non lo sono, ad esempio nel manuale di MySQL parole come "mysql" e "table" risultano utilizzate così di frequente che non aiutano nella ricerca di un particolare argomento.

La costruzione di un insieme di stopword deve essere fatta con cautela, infatti, la rimozione di troppe parole può peggiorare l'efficacia del motore di ricerca; per esempio alcune query valide potrebbero non dare alcun risultato.

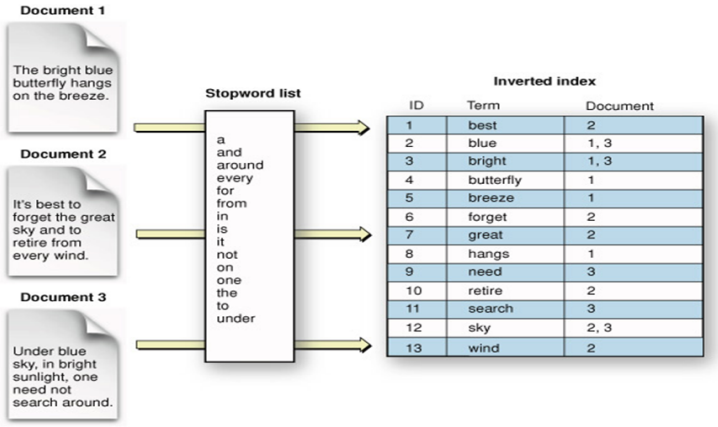

Figura 1. *Inverted index* e eliminazione delle *stopword*, rappresentati in modo schematico.

2.2 Reperimento

Come modello di reperimento si è utilizzato il modello probabilistico, paradigma dominante nell'implementazione di sistemi IR.

Questo modello si basa sul Probability Ranking Principle(PRP)[6] ovvero:

> "Se un Sistema IR risponde a ciascuna interrogazione con una lista di documenti ordinati in modo non crescente per probabilità di rilevanza all'esigenza informativa dell'utente che ha espresso l'interrogazione, e posto che le probabilità siano state stimate nel modo migliore possibile sulla base dei dati a disposizione, allora l'efficacia complessiva del sistema è la maggiore possibile sulla base di quei dati."

In questo modello i documenti sono rappresentati come vettori in cui ogni elemento, relativo ad un termine, assume valore 1 o 0 a seconda che il termine corrispondente sia presente nel documento o meno. Inoltre si fa l'assunzione che la presenza di un termine sia indipendente dalla presenza degli altri termini. Per questo si parla anche di *binary independence model*.

Nel contesto del modello probabilistico, lo schema di pesatura comunemente utilizzato nella funzione che assegna l'ordine ai documenti a seconda del grado di rilevanza è il cosiddetto BM25[1] (Best Matching 25).

La formula della funzione di scoring per il BM25 è:

$$\sum_{i \in Q} \left[log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)} \right) \cdot \frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i} \right]$$

Sommatoria su tutti i termini contenuti nella query Q; dove r_i è il numero di documenti rilevanti contenenti il termine i, n_i è il numero di documenti contenenti il termine i, N è il numero totale di documenti nella collezione R è il numero di documenti rilevanti per la query, f_i è la frequenza del termine i nel documento, qf_i è la frequenza del termine i nella query e k_1 , k_2, e K sono parametri i cui valori sono assegnati empiricamente.

Si ha inoltre la condizione che r_i e R sono posti a 0 nel caso non ci sia informazione rilevante, i termini 0.5 e 1 servono per evitare complicazioni in questi casi.

Nella prima parte:

$$log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)} \right),$$

il numeratore è il rapporto tra il numero di documenti rilevanti in cui il termine i appare ed il numero di documenti rilevanti in cui non appare. In pratica si tratta di un rapporto di verosimiglianza che dice quanto il termine i è "rilevante".

Il denominatore è lo stesso rapporto solo che per i documenti non rilevanti, quindi indica quanto il termine è "non rilevante".

Nella seconda parte:

$$\frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i},$$

k_1 e k_2 servono per dare più o meno peso alla frequenza del termine i nel documento e nella query rispettivamente.

Infine K serve per normalizzare la frequenza del termine nel documento per la lunghezza del documento stesso.

Nel nostro caso, avendo documenti strutturati, risulta preferibile una variante del BM25 denominata BM25F (Best Match 25 Model with Extension to Multiple Weighted Fields) che combina efficacemente informazioni da campi diversi.

3 Interfaccia web

Parte di questo progetto è stata la creazione di un'interfaccia web tramite cui un utente può interrogare l'indice della collezione. In particolare l'interfaccia è rivolta ad utenti che cercano articoli pertinenti ad un certo argomento, definito dalla query.

Per il server web è stato utilizzato un modulo di python dedicato allo scopo: web-py. Questo modulo permette di integrare facilmente funzioni di python in pagine HTML.

Per lo stile delle pagine html si è optato per l'utilizzo degli strumenti disponibili da bootstrap[6].

La funzione di reperimento utilizzata è essenzialmente la stessa usata per gli esperimenti, adattata in modo da accettare query interattive, inserite dagli utenti.

Si è deciso di limitare il numero di risultati proposti a 1000 articoli in quanto si è pensato che, se un articolo rilevante fosse in una posizione dopo la millesima, non verrebbe visto dall'utente che tende a controllare solo le prime pagine.

La pagina iniziale ha solo la barra per l'inserimento della query con un immagine suggestiva che fa da sfondo.

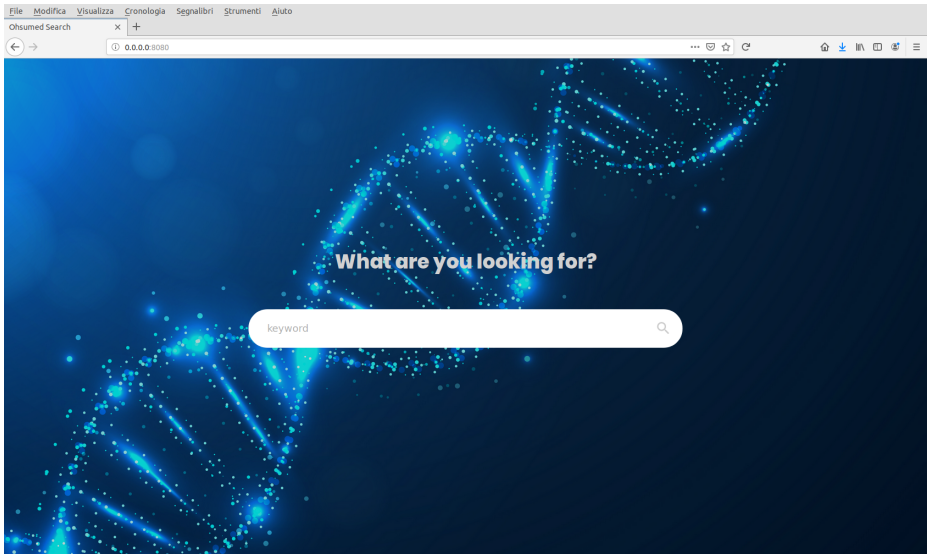

Figura 2. Pagina iniziale.

[6] https://getbootstrap.com/

Dopo aver effettuato il reperimento, l'utente viene indirizzato alle pagine contenenti i link dei risultati. Questi link poi rimandano alla pagina di *PubMed*[7] dove c'è l'articolo.

Se ne può vedere un esempio in Figura 3.

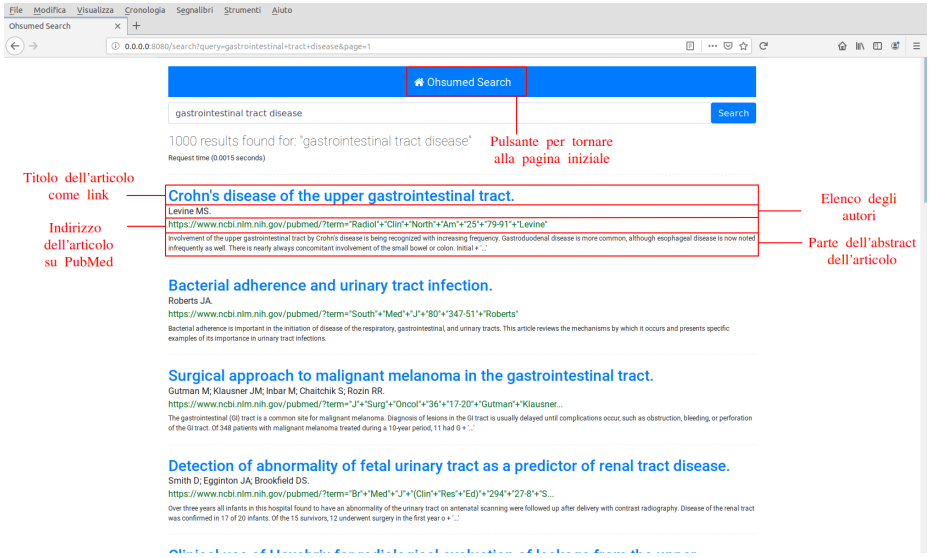

Figura 3. Pagina dei risultati.

È risultato utile sfruttare un modulo legato a whoosh (paginate-whoosh[8]) che dà la possibilità di ottenere facilmente la suddivisione dei risultati in pagine.

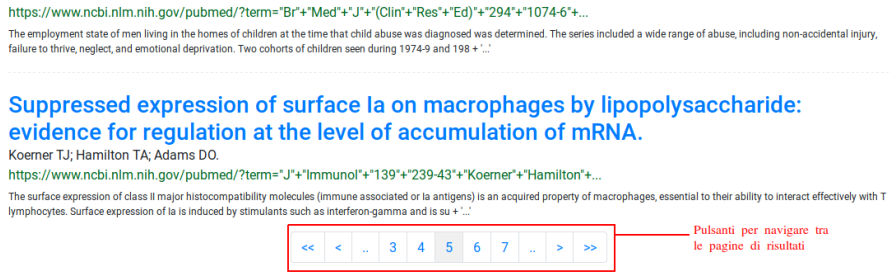

Figura 4. Pulsanti per navigare tra le pagine di risultati.

[7] https://www.ncbi.nlm.nih.gov/pubmed/
[8] https://pypi.org/project/paginate-whoosh/

Nel caso in cui non vengano trovati risultati si forniscono all'utente dei suggerimenti per correggere eventualmente la query.

Per come è scritta la funzione di reperimento, non sarebbe difficile implementare una funzionalità di "ricerca avanzata" dando all'utente la possibilità di cercare per autore o di scegliere l'operatore logico per collegare le parole della query.

4 Esperimenti e Risultati

Per gli esperimenti si è utilizzata parte della già citata collezione sperimentale chiamata OHSUMED[9].

Si tratta di una collezione di oltre 300'000 articoli provenienti dal database bibliografico MEDLINE pubblicati tra il 1987 ed il 1991; di questi documenti ne sono stati utilizzati 54'711.

I programmi utilizzati per effettuare indicizzazione e reperimento sono stati scritti in python(versione 2.7), sfruttando oggetti e funzioni del modulo whoosh[10].

Per la valutazione dei risultati ottenuti nei vari esperimenti si è invece utilizzato lo strumento standard trec_eval[11]. Questo programma permette di ottenere alcune misure della qualità di un sistema di information retrieval se si conoscono i documenti rilevanti per le query sperimentali utilizzate.

Come misura da usare nel confronto tra risultati di diverse run si è scelta la Mean Average Precision(MAP)[3] , quantità calcolabile sia a livello di singola query che a livello complessivo su un insieme di query. Inoltre, per verificare se i valori di MAP ottenuti in run diverse si discostano in modo statisticamente significativo gli uni dagli altri si effettuano test di Wilcoxon[4] su coppie di risultati.

4.1 Scelte iniziali

Prima di iniziare gli esperimenti sono state fatte delle scelte riguardo certi aspetti dei metodi di reperimento che rimanessero costanti durante tutti gli esperimenti. Queste scelte sono state fatte in seguito ad alcune run preliminari.

Come schema di pesatura, per la funzione di reperimento e ranking, si è deciso di utilizzare il BM25F, descritto nella sezione precedente, anche perchè si può considereare "state-of-the-art" dal punto di vista dei modelli per information retrieval su documenti strutturati, e si è deciso di utilizzare l'operatore OR come operatore logico per raggruppare le parole delle query, ovvero si cercano documenti in cui è presente almeno una parola della query.

Sempre a seguito delle run preliminari si è deciso di utilizzare il campo "desc" delle query anziché il campo "title" in quanto dà risultati migliori; inoltre, si è notato che in alcune query sperimentali sono presenti parole scritte in modo

[9] https://bit.ly/2wpOynZ
[10] https://whoosh.readthedocs.io/en/latest/index.html
[11] https://trec.nist.gov/trec_eval/

errato e che quindi non risultano presenti nell'indice. Per questo dopo aver verificato, leggendo i documenti rilevanti per quelle query, quali fossero gli errori si è ritenuto opportuno effettuare una correzione al momento del reperimento. A grandi linee, la correzione viene fatta su parole che non risultano presenti nell'indice cercando in quest'ultimo delle parole alternative che si discostano di al più una lettera dalla parola originale; queste parole vengono quindi aggiunte alla query di partenza e si procede con la ricerca.

Un'ultima scelta è stata fatta riguardo il numero massimo di documenti reperiti. Si è optato per 100 documenti poiché un numero maggiore non porta grandi miglioramenti dal punto di vista del *MAP*.

Affinché Whoosh possa indicizzare una collezione di documenti, necessita la specificazione di uno schema che include, per ogni possibile campo dei documenti della collezione, il nome del campo ed il tipo del campo.

Lo schema di base per la collezione qui utilizzata è il seguente:

```
schema = Schema(docid          = ID(stored=True),
                title          = TEXT(stored=True),
                identifier     = ID(stored=True),
                terms          = NGRAM(stored=True),
                authors        = NGRAM(stored=True),
                abstract       = TEXT(stored=True),
                publication    = TEXT(stored=True),
                source         = TEXT(stored=True))
```

Figura 5. Schema necessario all'indicizzazione dei documenti. "stored=True" indica che il campo viene salvato ed è quindi successivamente accessibile.

4.2 Baseline

I valori di *MAP* scelti come baseline sono quelli ottenuti effettuando ricerche su un indice con lo schema di base, riportato sopra, senza l'utilizzo di una lista di stopword e cercando in due campi dei documenti: "title" e "abstract". Cercando su un campo ("title") o tre campi ("title", "abstract" e "terms") risulta solo in un peggioramento dei risultati.

Si possono vedere i risultati ottenuti usando la configurazione "baseline" riassunti in Tabella 1.

un campo	**due campi**	**tre campi**
num*q all 63*	num*q all 63*	num*q all 63*
num*ret all 6228*	num*ret all 6244*	num*ret all 6271*
num*rel all 670*	num*rel all 670*	num*rel all 670*
num*rel*ret all 349	num*rel*ret all 419	num*rel*ret all 367
map all 0.2130	map all **0.2744**	map all 0.2155

Tabella 1. Risultati trec_eval complessivi per tutte le query, nessuna manipolazione del testo, numero risultati restituiti per ogni query = 100, pesatura BM25F.

Come si nota dalla tabella il valore baseline di *MAP* è 0.2744.

4.3 Esperimenti

Gli esperimenti sono consistiti in tre prove in cui si è cambiato l'insieme di stopword usato nell'indicizzazione. In particolare ad ogni prova si è utilizzato un insieme di stopword con parole in più rispetto a quello delle prove precedenti.

Come con la baseline, per ciascuna prova si è effettuata la ricerca per uno, due e tre campi.

Prima prova: Per la prima prova sono state utilizzate le stopword generali della lingua inglese. Queste comprendono termini come "the", "that", "is"; sono poi state aggiunte anche le lettere dell'alfabeto ed i numeri.

Come si vede in Tabella 2, non sembra si sia ottenuto un miglioramento significativo rispetto alla baseline. Il *MAP* ottenuto utilizzando due campi "migliora" solo dello 0.0001.

un campo	**due campi**	**tre campi**
num*q all 63*	num*q all 63*	num*q all 63*
num*ret all 6228*	num*ret all 6244*	num*ret all 6271*
num*rel all 670*	num*rel all 670*	num*rel all 670*
num*rel*ret all 350	num*rel*ret all 418	num*rel*ret all 367
map all 0.2137	map all **0.2745**	map all 0.2152

Tabella 2. Risultati trec_eval, rimozione delle stopword generali, numero risultati restituiti per ogni query = 100, pesatura BM25F.

Seconda prova: Nella seconda prova si è allora cercato di migliorare i risultati aggiungendo alle stopword generali, alcune parole che non sono normalmente considerate stopword ma che nel contesto medico, come nel caso della collezione OHSUMED, diventano tali.

Per questo abbiamo utilizzato un insieme di stopword denominate stopword_cliniche che oltre ad avere le stopword generali ha anche parole come "medical", "condition" o "family"[12].

Con questa prova sono stati ottenuti risultati accettabili, infatti come si vede in Tabella 3, l'aumento del *MAP* rispetto alla baseline è stato quasi dello 0.01:

un campo	**due campi**	**tre campi**
num*q all 63*	num*q all 63*	num*q all 63*
num*ret all 6228*	num*ret all 6244*	num*ret all 6271*
num*rel all 670*	num*rel all 670*	num*rel all 670*
num*rel*ret all 350	num*rel*ret all 419	num*rel*ret all 376
map all 0.2156	map all **0.2837**	map all 0.2166

Tabella 3. Risultati trec_eval, rimozione delle stopword cliniche, numero risultati restituiti per ogni query = 100, pesatura BM25F.

[12] https://github.com/kavgan/clinical-concepts/blob/master/clinical-stopwords.txt

Terza prova: Nell'ultima prova sono state aggiunte altre parole alla lista delle stopword. La scelta delle parole da aggiungere è stata fatta in due fasi principali.

Nella prima è stata ricavata la frequenza con cui le parole delle query appaiono nei rispettivi documenti rilevanti. Dopo di ciò le parole con frequenza prossima allo 0 sono state prese in considerazione come possibili stopword.

Nella seconda fase sono state prese le query che, basandosi sui risultati della seconda prova, hanno avuto un *MAP* relativamente basso(inferiore a 0.2). Sono state quindi prese alcune parole di queste query come altre possibili stopword anche basandosi sulla loro frequenza nei documenti rilevanti.

Per alcune parole la scelta se includerle nell'insieme di stopword è stata resa difficile dal fatto che, essendo presenti in più di una query la loro rimozione poteva significare il miglioramento di un *MAP* ed il peggioramento di un altro.

Per questo, prima di arrivare all'insieme finale sono stati fatti alcuni tentativi.

Come si può notare dalla Tabella 4 si ha un aumento del *MAP* rispetto alla baseline decisamente superiore alle prove precedenti, con circa lo 0.075 in più.

un campo	**due campi**	**tre campi**
num*q all 63*	num*q all 63*	num*q all 63*
num*ret all 5282*	num*ret all 5690*	num*ret all 5788*
num*rel all 670*	num*rel all 670*	num*rel all 670*
num*rel*ret all 386	num*rel*ret all 463	num*rel*ret all 435
map all 0.2720	map all **0.3508**	map all 0.2813

Tabella 4. Risultati trec_eval, rimozione delle stopword cliniche migliorate, numero risultati restituiti per ogni query = 100, pesatura BM25F.

La Figura 6 riassume i risultati delle tre prove e si può notare un netto miglioramento utilizzando le stopword migliorate.

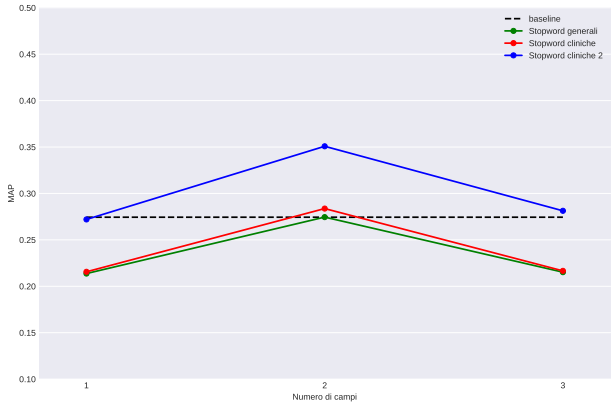

Figura 6. Variazione del *MAP* cambiando il numero di campi per i tre insiemi di stopword, la linea orizzontale indica la baseline di 0.2744.

4.4 Test di significatività :

Per avere conferma che le differenze tra i *MAP* ottenuti nelle varie prove e nella baseline siano significativamente diversi si è ritenuto opportuno effettuare dei test. Tra i test che si utilizzano comunemente a questo scopo si è scelto di utilizzare il test di Wilcoxon che a differenza dell'alternativo t-test risulta avere più potenza statistica in caso di non normalità dei dati.

I test sono stati fatti tra i *MAP* delle prove utilizzando due campi in quanto hanno dato generalmente risultati migliori.

L'ipotesi alternativa è unilaterale ed equivale a dire che, tra le due prove confrontate, la seconda prova ha *MAP* maggiore della prima. Per contro l'ipotesi nulla è che la prima prova ha *MAP* maggiore o uguale alla seconda.

Prove a confronto	**statistica test**	**p-value**
BASELINE contro STOP1	582	0.0393
BASELINE contro STOP2	551.5	0.0092
STOP1 contro STOP2	549.5	0.0056
BASELINE contro STOP3	276.0	1.2749e−6
STOP1 contro STOP3	291.0	1.2936e−6
STOP2 contro STOP3	290.0	3.5437e−6

Tabella 5. Statistiche test e livelli di significatività osservati per i test sulla differenza dei *MAP*. STOP1 indica le stopword generali, STOP2 indica le stopword cliniche e STOP3 indica le stopword cliniche migliorate.

Come si vede in Tabella 5 i miglioramenti portati dalle stopword delle prime due prove hanno p-value che permettono di rifiutare l'ipotesi nulla solo in caso si prenda come soglia il livello di significatività dello 0.05. Invece, con la terza prova il p-value risulta molto più basso portando a rifiutare l'ipotesi nulla in maniera più convincente.

5 Considerazioni finali

Come si è visto dai risultati delle prove, l'utilizzo di stopword inerenti al contesto della collezione, porta a reperire più documenti rilevanti nelle prime posizioni. Bisogna, però, tenere in considerazione che nella terza prova, quella che ha dato i risultati migliori, l'insieme delle stopword creato è strettamente legato alle query sperimentali. Per questo c'è la possibilità che i risultati ottenuti con query diverse siano peggiori.

Riferimenti bibliografici

1. W. Bruce Croft and Donald Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley, (2009), pp. 250-252
2. W. Bruce Croft and Donald Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley, (2009), pp. 90-91
3. W. Bruce Croft and Donald Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley, (2009), pp. 313
4. W. Bruce Croft and Donald Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley, (2009), pp. 325-330
5. W. Bruce Croft and Donald Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley, (2009), pp. 129-130
6. S.E. Robertson, (1997) "THE PROBABILITY RANKING PRINCIPLE IN IR", Journal of Documentation, Vol. 33 Issue: 4, pp.294-304, https://doi.org/10.1108/eb026647

(a) Alessandro Stefani

(b) Caterina Buranelli

(c) Cristi Gutu