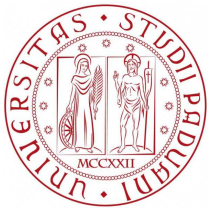


Università degli studi di Padova

Facoltà di Scienze Statistiche

Tesi di Laurea Magistrale in

Scienze Statistiche

4IR: Valutazione di un modello di Information Retrieval basato sulla trasformata discreta di Fourier

---

Candidato:
Mauro Brunazzo

Relatore:
Prof. Massimo Melucci

Correlatore:
Dr. Emanuele Di Buccio

Anno Accademico 2010/2011

SOMMARIO

In questa tesi viene sviluppato e sperimentato un nuovo modello di Information Retrieval (IR) denominato LSPR (*Least Spectrum Power Ranking*). A differenza dei modelli tradizionali, LSPR sfrutta la trasformata discreta di Fourier (*Discrete Fourier Transform*, DFT) per associare ad ogni termine dell'interrogazione un segnale sinusoidale e far in modo che l'interrogazione, risultato della somma di più segnali sinusoidali, sia analizzabile nel dominio delle frequenze. La base di partenza è descritta in [Costa, 2009] e [Costa and Melucci, 2010]; un'applicazione ai *recommender systems* dove LSPR da buoni risultati si può trovare in [Costa and Roda, 2011] .

Dal punto di vista metodologico, si è partiti dai fondamenti di LSPR per arrivare ad un'implementazione in Java, che impiega strutture dati efficienti, in modo da ottenere un motore di ricerca scalabile con il numero di documenti, utenti e interrogazioni. Il sistema di IR così sviluppato in forma prototipale prende il nome 4IR da ***Four****ier* ***I****nformation* ***R****etrieval* e rappresenta il primo contributo di questa tesi. La seconda parte del lavoro consiste nella valutazione di 4IR mediante l'utilizzo di una collezione sperimentale. Grazie al confronto con un modello rappresentativo dello stato dell'arte, si sono tratte alcune conclusioni sulle prestazioni e sui punti di forza e debolezza del modello.

INDICE

1 Introduzione **1**

1.1 Concetti di base dell'Information Retrieval 1

1.2 Scopi e obiettivi della tesi 3

1.3 Risultati e sviluppi futuri 4

2 Trasformata di Fourier e modello LSPR **5**

2.1 Trasformata di Fourier 5

2.1.1 Dispersione spettrale nella DFT 6

2.1.2 Teorema fondamentale del campionamento 7

2.1.3 Filtraggio dei segnali 7

2.2 Metafora del modello LSPR 8

2.3 Schemi di pesatura dei termini 10

3 Progetto e implementazione del sistema di Information Retrieval 4IR basato su LSPR **13**

3.1 Obiettivi e requisiti 13

3.2 Architettura generale 14

3.3 Progetto del motore di ricerca 15

3.3.1 Indicizzazione 15

3.3.2 Trasformazione interrogazione-spettro 15

3.3.2.1 Numero di campioni della DFT 16

3.3.2.2 Frequenza associata ad ogni termine dell'interrogazione 17

3.3.2.3 Calcolo dello spettro 17

3.3.3 Trasformazione documento-filtro 17

3.3.4 *Ranking* 19

3.4 Implementazione del motore di ricerca 19

3.4.1 Metodi principali 19

3.4.1.1 Analisi dell'interrogazione 20
3.4.1.2 Calcolo dello spettro associato all'interrogazione 20
3.4.1.3 Creazione dei filtri e filtraggio 22
3.4.2 Funzionamento di 4IR 24

4 Esperimenti **27**
4.1 Misure 27
4.2 Collezione sperimentale 31
4.3 Metodologia sperimentale 33
4.3.1 Baseline BM25 34
4.3.2 LSPR 35
4.4 Risultati sperimentali 35
4.4.1 Test su *title* 36
4.4.2 Test su *description* 37
4.4.3 Analisi di significatività 40
4.5 Studio di un *topic* "problematico" 43

5 Conclusioni **49**

Bibliografia **51**

CAPITOLO

UNO

INTRODUZIONE

1.1 Concetti di base dell'Information Retrieval

Fin dall'antichità l'essere umano ha cercato di soddisfare in vari modi un'esigenza insita nella sua natura, ossia il desiderio di conoscenza. Per anni la consultazione di libri ed enciclopedie ha appagato la curiosità e le necessità degli individui, ma l'avvento di Internet ha modificato il modo con cui si cercano informazioni. Ogni utente può, con un semplice *click*, entrare in un mondo che lui stesso può arricchire e modificare, un mondo che presenta una mole di dati esorbitante, di cui, però, solo una parte esigua soddisfa la sua esigenza informativa.

In questo scenario si è inserita l'Information Retrieval (IR), la scienza che studia le tecniche per estrarre informazioni utili da grandi masse di documenti. Questa disciplina ha avuto origine intorno agli anni '50 ma ha visto un grande sviluppo con la nascita dei motori di ricerca per il World Wide Web (WWW), i quali hanno proiettato l'utente verso una dimensione informativa che fino a qualche anno prima era del tutto impensabile. Un concetto fondamentale su cui si fonda l'IR è quello di collezione, ossia una raccolta di documenti che è di interesse rappresentare, descrivere e memorizzare; un utente che accede ad un sistema di IR ha tipicamente un'esigenza informativa da soddisfare e per risolvere il suo problema egli interroga il motore al fine di reperire, fra tutti i documenti della collezione, quelli che possano contenere informazioni per lui utili.

Il concetto di utilità o rilevanza dei documenti rispetto alle esigenze informative degli utenti non ha una definizione precisa ed è influenzato da molti fattori come ad esempio il contesto dell'interrogazione, l'esperienza di chi effettua la ricerca e i documenti che sono stati recuperati. Per esempio, un documento che è stato giudicato rilevante da un determinato utente, potrebbe non esserlo per un altro che possiede

conoscenze diverse. Da qui si intuisce una prima difficoltà intrinseca nello studio e lo sviluppo dell'IR: non è possibile conoscere in modo assoluto l'utilità o la rilevanza di un documento, ma questa deve essere sempre considerata in relazione al contesto. A questo punto è necessario introdurre un secondo problema con cui l'IR si scontra: le informazioni utili all'utente sono contenute in pochi, anzi pochissimi documenti; per questo motivo sarà necessario recuperare tutte e sole le informazioni di interesse tralasciando quelle non importanti e ciò non è certo di facile soluzione.

Un passaggio fondamentale nell'IR è l'indicizzazione, cioè, il processo che, dato un insieme di documenti, descrive l'informazione in essi contenuta; tutto ciò avviene estraendo da questi ultimi dei descrittori, ossia dati che esprimono gli aspetti salienti del contenuto informativo del documento in esame e che sono organizzati in uno o più indici. L'indice in un sistema di IR risiede nella memoria del calcolatore in cui è in esecuzione il motore di ricerca ed associa, ad ogni descrittore, i documenti che lo contengono. Da questo, si può capire come l'indice per un sistema di IR svolga un ruolo estremamente importante perché esso è l'unico mezzo per accedere ai documenti della collezione: tutto ciò che viene restituito dal sistema in risposta ad un'interrogazione è, pertanto, necessariamente indicizzato.

A seguito dell'indicizzazione il sistema di IR, mediante un algoritmo di reperimento e ordinamento, restituisce all'utente i documenti reperiti dall'indice in ordine decrescente di rilevanza. Posto che la rilevanza di un documento non sia determinabile univocamente dal sistema di IR, per quanto già detto prima, il compito dell'algoritmo di reperimento è quello di decidere, data un'interrogazione e un documento, quanto quest'ultimo sia utile a soddisfare l'esigenza informativa dell'utente e quindi di stabilire la posizione che dovrà occupare nella lista finale dei risultati.

Nell'ambito dell'IR, esistono diversi modelli teorici il cui scopo è proprio quello di formalizzare la rappresentazione di documenti e interrogazioni nonché di stabilire un indicatore di rilevanza di un documento rispetto all'interrogazione definita dall'utente. Il primo modello proposto fu quello booleano, forse il più semplice tra i modelli in IR, in quanto fa uso di una funzione di corrispondenza esatta tra documenti e interrogazione, dove, sia quest'ultima che il recupero sono basati sull'algebra booleana: ciò che viene restituito dal sistema sono i documenti che rendono vera l'interrogazione. Successivamente fu introdotto il modello vettoriale in cui interrogazione e documenti sono rappresentati come vettori con una componente per ogni descrittore e il cui valore rappresenta il peso del descrittore corrispondente; un documento è giudicato tanto più importante quanto più sono vicini i vettori che rappresentano l'interrogazione e il documento stesso. Infine, nel modello probabilistico, i descrittori sono modellati come variabili casuali e i documenti sono rappresentati come l'esito della realizzazione di una variabile aleatoria; il reperimento è basato sulla probabilità di rilevanza di un

documento data un'esigenza informativa e i documenti sono ordinati per probabilità di rilevanza decrescente.

Posto ora di aver a disposizione un sistema di IR, è di notevole interesse avere delle misure che diano un'indicazione sulle *performance* del sistema; questo può essere utile sia per confrontare il sistema con un altro sia per paragonare due diverse versioni del sistema stesso. Questa fase prende il nome di valutazione, durante la quale si devono tenere presenti due proprietà principali dei sistemi di IR: l'efficacia, relativa ai documenti rilevanti e non rilevanti reperiti, e l'efficienza, relativa alla velocità, all'utilizzo di risorse computazionali, alla quantità di memoria occupata dal sistema e alla banda di rete utilizzata.

Si può dunque intuire come un sistema di IR sia una struttura complessa, formata da più componenti che interagiscono tra di loro; solo la corretta analisi e progettazione di ogni singola componente e del loro insieme può condurre verso un sistema stabile e scalabile nel numero di interrogazioni,utenti e documenti.

1.2 Scopi e obiettivi della tesi

La tesi ha l'obiettivo di implementare e valutare un nuovo modello di IR, presentato in [Costa and Melucci, 2010] e basato sul concetto di serie di Fourier. In quel lavoro si è affrontato il problema teorico e ci si è limitati ad una prima conferma sperimentale: dagli esperimenti effettuati su una collezione di qualche migliaio di documenti, il modello ha dimostrato di avere caratteristiche di efficacia confrontabili con quelle dei modelli allo stato dell'arte.

Lo scopo di questa tesi è stato quindi, dato il modello teorico trattato nel lavoro citato pocanzi, procedere con la sua implementazione in Java tenendo conto dell'ottimizzazione del codice e utilizzando costrutti e strutture di dati efficienti; in questo modo ci si è concentrati, oltre che sull'efficacia, anche sull'efficienza del sistema di IR ponendo, tra gli obiettivi, il miglioramento dei tempi di esecuzione e dello spazio in memoria occupato dall'indice. Un altro aspetto a cui ci si è dedicati durante la fase di implementazione è stata la variazione di alcuni parametri dell'algoritmo verificando come cambiava la risposta del modello al variare di questi parametri e cercando quindi di trovare dei valori ottimali; questa parte sarà trattata in modo approfondito nel capitolo 4.

Una volta realizzato il sistema di LSPR, si è proceduto alla valutazione su una collezione composta da quasi un milione e settecentomila documenti, al fine di verificare come il modello si comporta in presenza di un numero elevato di dati e di constatare, quindi, se l'efficacia riscontrata in [Costa, 2009] su una collezione ridotta sia confermata anche nel caso in cui il sistema operi su un insieme più vasto. Poiché le prestazioni

si sono confermate buone anche con collezioni più grandi, si può pensare di utilizzare questo modello in altri ambiti, come in quello dei motori di ricerca per il WWW, delle reti P2P (*peer-to-peer*), o dei *desktop*. Inoltre, è possibile combinarlo con altri modelli già esistenti o applicarlo per compiti diversi.

A seguito dello studio sui concetti teorici del nuovo modello LSPR, si è proceduto alla sua realizzazione in Java utilizzando come API Lucene, una libreria *open source* che fornisce le funzionalità di base di IR; maggiori dettagli a riguardo si possono trovare in [Hatcher et al., 2010]. L'algoritmo sviluppato consiste in una serie di moduli, ognuno dei quali implementa una particolare funzionalità richiesta dal modello. Una volta realizzato il software, si è utilizzata per la fase di test una collezione sperimentale [1] TREC, in particolare la "*2001 web trac ad hoc topics*" i cui dettagli verranno presentati nella sezione 4.2.

1.3 Risultati e sviluppi futuri

I risultati finali ottenuti evidenziano che il sistema 4IR risulta stabile e scalabile con il numero di documenti e interrogazioni, quindi l'obiettivo che ci si era posti ad inizio lavoro è stato raggiunto. La misurazione della scalabilità con il numero di utenti è vincolata dal carattere laboratoriale degli esperimenti. Inoltre, sono stati messi in luce alcuni punti di forza del modello LSPR, in particolare emerge una buona capacità di reperire documenti rilevanti nel caso in cui l'interrogazione posta al sistema contenga un numero ristretto di termini; questa caratteristica rende il modello LSPR adatto ad operare in ambienti dove l'esigenza informativa è piuttosto breve. Dato che, in contesti come la ricerca nei motori per il WWW, le interrogazioni inviate ad un motore di ricerca sono per la maggior parte composte da un numero ridotto di termini, LSPR appare un valido modello da poter applicare negli ambiti in seguito suggeriti a titolo d'esempio nel capitolo 5.

[1] Insieme di documenti, interrogazioni e giudizi di rilevanza per alcuni documenti, data un'interrogazione.

CAPITOLO

DUE

TRASFORMATA DI FOURIER E MODELLO LSPR

In questo capitolo vengono richiamati alcuni concetti teorici alla base del modello LSPR. La trattazione non può essere esaustiva e perciò si rimanda a [Briggs et al., 1995] o [Dym and McKean, 1972] per i dettagli.

2.1 Trasformata di Fourier

Un segnale x(t) è detto periodico quando si ripete identicamente dopo un intervallo di tempo T, detto periodo; invertendo il periodo si ottiene la frequenza del segnale:

$$f = \frac{1}{T} \tag{2.1}$$

La serie di Fourier, introdotta dal matematico Jean Baptiste Joseph Fourier, permette di rappresentare un segnale periodico mediante somma di funzioni sinusoidali. Grazie alla serie di Fourier è possibile scomporre segnali complicati in somma di funzioni più semplici rendendone agevole l'analisi.

Nel caso in cui il segnale da analizzare sia continuo e non periodico, è stato introdotto un metodo per poter procedere in ogni caso alla sua analisi e questo viene fatto mediante la trasformata di Fourier; infatti, un segnale non periodico può essere visto come un segnale periodico di periodo infinito. Se consideriamo il caso in cui il segnale in ingresso è discreto, bisogna ricorrere alla trasformata discreta di Fourier (*Discrete Fourier Transform*, DFT) che permette di effettuare l'analisi in frequenza di segnali a tempo discreto; la DFT è quindi una trasformata per l'analisi di Fourier di funzioni su un dominio limitato e discreto. In particolare, la DFT richiede in ingresso una sequenza finita di numeri reali o complessi, che rendono la trasformata ideale per l'elaborazione di informazioni su un elaboratore elettronico; i valori in ingresso sono

spesso ottenuti per campionamento di una funzione continua, ad esempio, la voce di una persona.

L'interesse per la DFT in IR deriva dal fatto che i segnali con cui si rappresentano i termini dell'interrogazione sono essenzialmente segnali a tempo discreto periodici [Costa and Melucci, 2010].

2.1.1 Dispersione spettrale nella DFT

Uno dei maggiori problemi della DFT è che, se la frequenza f_0 del segnale non è un multiplo intero del quanto di frequenza F [1], quella che si va ad ottenere è una DFT distorta dalla presenza di componenti non nulle in un intorno di f_0 (figura 2.2). Nel caso di un segnale sinusoidale come quello analizzato in questa tesi, la DFT corretta sarebbe un impulso alla frequenza f_0 del segnale, come rappresentato in figura 2.1 . Il fenomeno che si vede nella figura 2.2 è noto in letteratura con il nome di dispersione spettrale e rappresenta un problema nell'analisi dello spettro del segnale. In [Costa and Melucci, 2010], invece, la dispersione spettrale viene usata come un vantaggio in quanto la presenza di componenti "attive" interviene quando nel documento è presente un termine x_{ij} che non è esattamente quello presente nell'interrogazione q_t ma ha una frequenza vicina. In questo modo si ottiene che il termine x_{ij} del documento, seppur non coincidendo con il termine presente nell'esigenza informativa, fa diminuire un pò la potenza associata all'interrogazione poiché nelle frequenze vicine a quella corrispondente a q_t, il modulo della DFT non è nullo. Per ottenere il fenomeno della dispersione spettrale, bisogna introdurre alcuni vincoli nella scelta di F e di f_i; questo sarà chiarito con maggior dettaglio nella sezione 3.3.2.1.

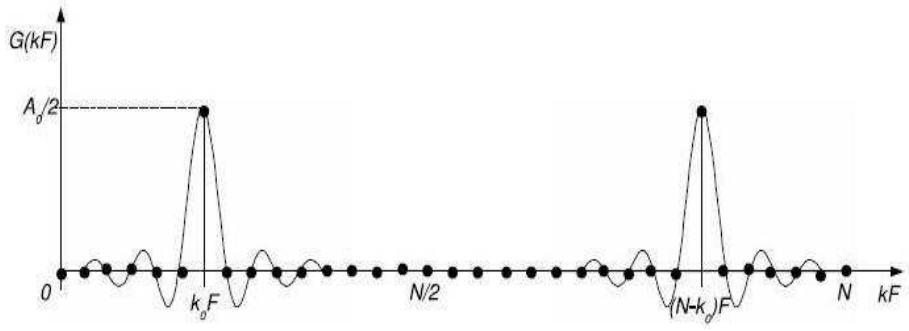


Figura 2.1: DFT di un segnale sinusoidale con f_i multipla di F

[1] Distanza tra due campioni successivi della trasformata di un segnale a tempo discreto.

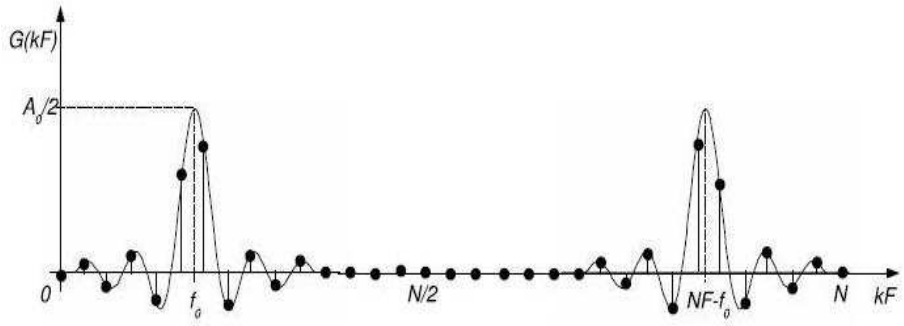


Figura 2.2: DFT di un segnale sinusoidale con f_i non multipla di F

2.1.2 Teorema fondamentale del campionamento

In elettronica, quando abbiamo a disposizione un segnale analogico, per poter rappresentare tale segnale in digitale si procede al suo campionamento; esso consiste nel prelevare il valore del segnale analogico in momenti scelti ad ogni intervallo di tempo T. Una domanda che è lecito porsi è se, sotto opportune condizioni, sia possibile ricostruire il segnale di partenza dai suoi campioni ovvero a che frequenza devono essere effettuati i campionamenti del segnale analogico. Intuitivamente, la frequenza di campionamento dipenderà dalla "variabilità" del segnale; infatti, più un segnale varia velocemente, più fittamente bisognerà campionarlo per riuscire a non perdere informazione. Questa idea, trova una formalizzazione nel teorema del campionamento di Nyquist-Shannon [Oppenheim et al., 1996] il quale afferma che, per poter risalire da un segnale campionato all'originale senza distorsione, la frequenza di campionamento deve essere pari almeno al doppio della frequenza massima del segnale. Formalizzando, detta F_c la frequenza di campionamento e f_M la frequenza più alta del segnale analogico, in base a quanto detto, per poter ricostruire il segnale analogico senza perdita di informazioni deve valere la relazione:

$$F_c \geq 2 \cdot f_M \tag{2.2}$$

2.1.3 Filtraggio dei segnali

Il termine **filtro** è usato per descrivere un dispositivo che discrimina, in accordo ad alcune caratteristiche degli oggetti applicati in ingresso, cosa può passare attraverso di esso e cosa no.

Nell'ambito di questa tesi verranno trattati solo filtri di tipo digitale, cioè quelli che permettono di compiere alcune funzioni matematiche su dei campioni di segnali

discreti nel tempo per aumentare o ridurre alcuni aspetti del segnale analizzato. In sostanza, quindi, un filtro digitale riceve in ingresso un segnale $x(nT)$ e restituisce in uscita un diverso segnale $y(nT)$. Se applichiamo la DFT ad entrambi i segnali e al filtro si ottiene la relazione:

$$\tilde{Y}(f) = \tilde{H}(f) \cdot \tilde{X}(f) \tag{2.3}$$

dove $\tilde{X}(f)$, $\tilde{Y}(f)$ e $\tilde{H}(f)$ indicano rispettivamente la DFT del segnale di uscita, la DFT del segnale di uscita e la DFT del filtro. Da questa relazione si nota che l'operazione di filtraggio si riduce ad una semplice moltiplicazione tra due funzioni.

Esistono molti tipi di filtri digitali che possono essere utilizzati; nell'ambito di questo lavoro, per effettuare l'operazione di filtraggio, è stato utilizzato qualcosa simile a un filtro *notch*, chiamato anche "filtro elimina banda" in quanto elimina lo spettro nell'intorno ad una frequenza predefinita mentre mantiene quasi inalterato lo spettro in corrispondenza di altri valori di frequenze. Per essere precisi, nella tesi, il filtro che si è utilizzato è un'approssimazione di un filtro *notch*, presenta una forma triangolare ed è possibile visualizzarne un'illustrazione grafica nella figura 3.3 della sezione 3.3.3 dove si spiegherà in dettaglio l'operazione di filtraggio.

2.2 Metafora del modello LSPR

Nell'ambito dell'IR, quando si deve presentare il funzionamento di un modello teorico, si utilizza una metafora. Essa ha lo scopo di sostituire ad un termine proprio un altro legato al primo da un rapporto di somiglianza cercando in questo modo di spiegare, a livello intuitivo, il modo d'utilizzo del modello. In questo paragrafo verrà presentato il funzionamento del modello di reperimento LSPR mediante la sua metafora.

Supponiamo che una persona debba recuperare un oggetto e che, per raggiungerlo, abbia a disposizione un percorso Q. Questo percorso ha la caratteristica d'essere attraversato da radiazioni che differiscono sia in intensità che in frequenza. Indicando con |Q| il numero totale di radiazioni presenti nel percorso, ci si riferirà all'i-esima radiazione indicandola con q_i; inoltre, l'ampiezza e la frequenza di questa radiazione verranno chiamate rispettivamente f_i e A_i.

Si stabilisce inoltre che la persona che deve recuperare l'oggetto abbia a disposizione |C| tute che proteggono dalle radiazioni, ognuna caratterizzata dalla presenza di componenti che riescono ad attenuare, in maniera diversa, le radiazioni associate a certe frequenze. Chiamata X_j la tuta j-esima, si suppone che essa contenga $|X_j|$ moduli; si può quindi caratterizzarla come $X_j = \{x_{1j}, x_{2j}, \ldots, x_{|X_j|j}\}$ dove ogni modulo x_{ij} è definito da una frequenza f_i e da una capacità di bloccare le radiazioni $\hat{w}_{ij}$. Una caratteristica importante della tuta è che essa, oltre a bloccare completamente $|X_j|$

tipologie di radiazioni, riesce ad attenuare radiazioni con frequenze vicine a quella che blocca.

La persona che deve recuperare l'oggetto, quindi, dovrà scegliere la tuta più adatta per affrontare il percorso, cioè quella che riesce a bloccare meglio le radiazioni nel tragitto in modo tale da non subire danni durante il percorso.

Da quanto spiegato finora, effettuare il passaggio dalla metafora al modello di IR dovrebbe risultare abbastanza facile. Le tute rappresentano i documenti della collezione, mentre i vari moduli che le compongono sono i descrittori dei documenti. Ogni termine è quindi caratterizzato da una frequenza e da una capacità di bloccare la radiazione che in concreto sarà il peso del termine nel documento.

Il percorso che la persona deve effettuare è invece rappresentato dall'interrogazione e le varie radiazioni q_i sono i suoi termini; la scelta della tuta più adatta per il percorso, ossia dei documenti più adatti all'interrogazione è in pratica il *ranking*.

Rimane adesso solo un ultimo dettaglio da chiarire: come si riesce a decidere quanto un documento è buono rispetto all'interrogazione? Per compiere questa scelta ci viene in aiuto la DFT; infatti, ottenuti i valori di intensità e frequenza associati ai vari termini dell'interrogazione, essi sono usati per definire dei segnali sinusoidali in modo che al termine q_i, caratterizzato da intensità A_i e frequenza f_i, corrisponda il segnale sinusoidale $A_i \sin(2\pi f_i nT)$. Il modulo della DFT di tale segnale assume valore massimo intorno a f_i per poi diminuire gradualmente (fenomeno della dispersione spettrale trattato nella sezione 2.1.1). Da questo si intuisce che ogni interrogazione avrà uno spettro ad essa associato con dei picchi vicino alle frequenze f_i dei termini che la compongono. Per avere un'indicazione sulla bontà di un documento rispetto all'esigenza informativa dell'utente è sufficiente filtrare lo spettro dell'interrogazione con dei filtri le cui caratteristiche dipendono da $\hat{w}_{ij}$; fatto ciò si procede al calcolo della potenza. [2]

Infine, per ottenere il *ranking*, si procede ordinando i documenti per potenza crescente; infatti, se un documento X_i, dopo la fase di filtraggio, fornisce una potenza maggiore rispetto a quella ottenuta attraverso il filtraggio del documento X_j, allora il secondo è da preferire al primo rispetto all'interrogazione posta. Tornando alla metafora del modello, questo sta a significare che le radiazioni lungo il percorso Q sono attenuate meno dalla tuta X_i che dalla tuta X_j, per cui è da preferire la seconda alla prima. Nella fase di *ranking* troveremo quindi prima il documento X_j e poi il documento X_i

[2] In questo lavoro, data la simmetria dello spettro, con potenza si intende la somma dei moduli di metà spettro.

2.3 Schemi di pesatura dei termini

Nell'ambito dell'IR, ci si ritrova molto spesso ad operare con il peso che un termine ha in un documento. A questo proposito si possono utilizzare misure diverse, a seconda dello scopo che si deve perseguire. Assumiamo di voler calcolare il peso che un termine i ha in un documento j e indichiamo tale peso con w_{ij}; un primo modo di procedere può essere quello di assegnare al termine peso 1 se è presente nel documento, peso 0 altrimenti. Nel caso appena presentato, il sistema di pesi risulta procedere secondo una logica binaria; esso, tenendo conto solamente della presenza/assenza del descrittore in un documento, appiattisce l'importanza dei descrittori che caratterizzano il contenuto del documento in quanto assegna ugual importanza sia ad un documento in cui il temine appare 1 volta che ad un altro in cui lo stesso termine appare n volte. Per tener conto anche della frequenza di apparizione di un termine è stata introdotta la misura TF(*Term Frequency*) definita in questo modo:

$$w_{ij} = f_{ij} \tag{2.4}$$

dove f_{ij} rappresenta il numero di occorrenze di i in j. Resta il problema che, se un termine di un'interrogazione appare in quasi tutti i documenti della collezione, esso dovrebbe avere importanza minore rispetto ad un termine che appare in meno documenti; a questo proposito è stato introdotto lo schema di pesatura IDF (*Inverse Document Frequency*) definito nel modo seguente:

$$w_{ij} = \begin{cases} \log_2\left(\frac{|C|+0.5}{n_i+0.5}\right) & n_i > 0 \\ 0 & n_i = 0 \end{cases} \tag{2.5}$$

dove |C| rappresenta il numero totale di documenti della collezione mentre n_i indica il numero di documenti della collezione contenenti il termine i. Infine, lo schema di pesatura TF-IDF considera entrambi i contributi e quindi si ha che:

$$w_{ij} = f_{ij} \cdot \log_2\left(\frac{|C|+0.5}{n_i+0.5}\right) \tag{2.6}$$

Nello sviluppo del modello LSPR è stato utilizzato lo schema di pesatura IDF e una variante di TF-IDF, denominata BM25, la cui funzione di ordinamento computa un peso tra il documento e l'interrogazione basandosi sulla probabilità dei termini dell'esigenza informativa di apparire nel documento; più alta sarà questa probabilità, maggiore sarà la relazione tra interrogazione e documento.

Per entrare più nel dettaglio quello che viene usato è il principio di ordinamento per probabilità (*Probability Ranking Principle*, PRP). L'idea alla base è che, considerando l'impossibilità per un sistema di IR di conoscere i valori di rilevanza per ogni documento e per ogni interrogazione, si può sfruttare un'informazione di tipo probabilistico. Posto

infatti di poter stimare con i dati a disposizione e nel modo migliore possibile una probabilità di rilevanza per ogni documento, data una specifica esigenza informativa, il PRP stabilisce che:

> Se un sistema di IR, con i dati disponibili, ordina i documenti reperiti in risposta ad un'interrogazione per probabilità decrescente di rilevanza stimata, allora l'efficacia complessiva del sistema è la migliore che si possa ottenere con i dati a disposizione.

Resta quindi da definire come viene calcolato il peso di un documento data un'interrogazione o, in altre parole, la probabilità di rilevanza di un documento. Partendo dal principio di ordinamento per probabilità appena citato e mediante semplici passaggi algebrici trattati in dettaglio in [Robertson and Zaragoza, 2009] è possibile dimostrare che la probabilità di rilevanza di un documento per una certa esigenza informativa si ottiene sommando i pesi di ogni termine dell'interrogazione presente nel documento. L'unica assunzione che è necessario fare per arrivare ad ottenere la funzione peso è un'assunzione di indipendenza condizionata; si ipotizza infatti che i termini dell'interrogazione siano indipendenti condizionatamente all'evento di rilevanza e a quello di non rilevanza. Indicando quindi con rel l'evento rilevanza, d un documento e q un'interrogazione, possiamo riassumere quanto detto con il seguente risultato algebrico:

$$P(\text{rel}|d,q) = \sum_{q, TF_i > 0} \omega_i \tag{2.7}$$

dove ω_i rappresenta il peso del termine i-esimo dell'interrogazione nel documento d, mentre TF_i al solito indica il numero di occorrenze del termine i in d. Il motivo del vincolo $TF_i > 0$ nella 2.7 è di natura computazionale: vengono sommati i pesi dei soli termini dell'interrogazione che compaiono nel documento d; se un termine dell'interrogazione non è presente in d, allora il suo peso è nullo.

Per poter quindi arrivare ad ottenere il peso del documento d condizionatamente ad un'esigenza informativa è necessario stabilire come vengono calcolati i singoli pesi ω_i e introdurre quindi la seguente notazione:

C — Collezione di documenti
|C| — Numero di documenti in C
n_i — Numero di documenti in C contenenti il termine i
R — Numero di documenti rilevanti
r_i — Numero di documenti rilevanti contenenti il termine i
dl — Lunghezza del documento d
$avdl$ — Lunghezza media dei documenti in C

La lunghezza del documento, dl, è ottenuta semplicemente sommando le frequenze di tutti i termini che compongono il documento stesso:

$$dl = \sum_{i \in d} TF_i \tag{2.8}$$

mentre la lunghezza media dei documenti in C è ricavata prendendo il numero totale di termini presenti nella collezione e dividendolo per il numero di documenti in C.

Si hanno ora a disposizione tutti gli ingredienti necessari per poter presentare la formula dei pesi BM25:

$$w_i^{BM25} = \frac{TF}{k_1 \left((1-b) + b\frac{dl}{avdl}\right) + TF} \cdot \log\left(\frac{(r_i + 0.5)(|C| - R + r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)}\right) \tag{2.9}$$

nella 2.9 b e k_1 sono due costanti che è necessario settare; da prove sperimentali, viene consigliato di scegliere i due parametri tenendo conto dei vincoli $0.5 \leq b \leq 0.8$ e $1.2 \leq k_1 \leq 2$.

Una perplessità che può sorgere dallo sguardo alla formulazione di BM25 è l'uso dell'informazione sui documenti rilevanti ad un'interrogazione, contenuta in R e r_i. Com'è possibile avere informazioni sui documenti rilevanti se, come più volte ribadito, un concetto chiave dell'IR è proprio l'ignoranza dei documenti rilevanti ad una certa esigenza dell'utente? La risposta è semplice: il modello BM25 può essere usato sia in presenza che in assenza di informazioni sulla rilevanza. Qualora non sia disponibile alcun indizio sulla rilevanza dei documenti, sarà sufficiente porre il valore di R e r_i a 0; in questo caso, il secondo fattore della moltiplicazione nella 2.9 si riduce alla formulazione di IDF. La formulazione di BM25 risulta, nel caso non si abbiano informazioni sulla rilevanza, un prodotto tra due componenti la prima basata su TF e la seconda basata su IDF.

A questo punto il funzionamento di BM25 dovrebbe risultare chiaro: data un'interrogazione q e un documento d, si applica la formula 2.7 e si trova così la probabilità di rilevanza per d data q. Questa operazione viene ripetuta per tutti i documenti della collezione C e alla fine sia avrà una lista di probabilità, una per ogni documento di C; sarà quindi sufficiente, in base al PRP, ordinare i documenti per probabilità decrescente e si otterrà il *ranking* del modello BM25.

CAPITOLO

TRE

PROGETTO E IMPLEMENTAZIONE DEL SISTEMA DI INFORMATION RETRIEVAL 4IR BASATO SU LSPR

3.1 Obiettivi e requisiti

Come accennato nel capitolo 1, lo sviluppo di 4IR ha proceduto tenendo sempre presente gli obiettivi posti a inizio progetto. Uno fra questi era l'incrementalità dell'indice; si voleva, infatti, che il nuovo motore fosse in grado di indicizzare i documenti della collezione e mantenere l'indice aggiornato utilizzando il minor numero di risorse computazionali possibili. Un altro aspetto molto importante che il nuovo motore di ricerca doveva possedere era la scalabilità dell'indicizzazione, del reperimento e dell'ordinamento. Nella precedente implementazione dell'algoritmo, infatti, si erano utilizzate delle strutture dati che rendevano il flusso del prototipo e la complessità computazionale piuttosto elevati. I tipi di dati utilizzati per il prototipo precedente, inoltre, risiedevano completamente in memoria centrale, con la conseguenza che, finché LSPR lavorava in una collezione ristretta di documenti gli indici di efficienza erano piuttosto soddisfacenti, quando però si aumentava la dimensione dei dati su cui l'algoritmo operava, si riscontravano problemi di gestione di memoria. Da qui, dunque, è iniziato il lavoro di questa tesi che mirava a progettare e sviluppare un prototipo scalabile. Quello che è stato fatto, in sostanza, è stato, dati i costrutti di base teorici del modello, tradurre gli stessi in modo da poterli rappresentare in un calcolatore utilizzando delle strutture di dati efficienti. Di conseguenza, le operazioni di reperimento e ordinamento dell'algoritmo sono state ottimizzate fino ad ottenere tempi e occupazione di memoria minori rispetto di qualche ordine di grandezza rispetto a quelli della precedente versione del prototipo. Inoltre, dai test effettuati su larga scala, 4IR si è dimostrato un software stabile e performante, segno che l'implementazione attuale è adatta alla sua

applicazione su una vasta collezione di documenti.

3.2 Architettura generale

La fase di sviluppo di 4IR basato sulla trasformata discreta di Fourier ha visto, come già detto, l'utilizzo di alcune primitive di IR, che forniscono le funzionalità per l'indicizzazione e il reperimento delle informazioni. La figura 3.1 riassume, in modo schematico, l'ambiente di lavoro. Si nota che sono presenti due livelli distinti, separati

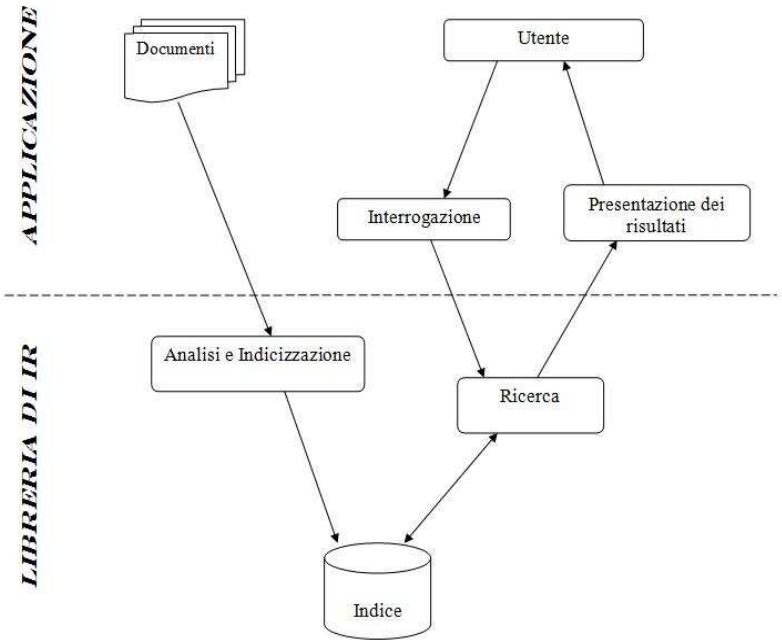


Figura 3.1: Ambiente di lavoro

nella figura da una linea orizzontale tratteggiata. Nella parte superiore incontriamo il livello di applicazione, quello direttamente a contatto con l'utente e con cui egli interagisce; è qui infatti che si trovano le maschere per inserire le interrogazioni e la presentazione dei risultati a video. Quello che l'utente non vede, ma che ricopre un ruolo fondamentale, sono le librerie di IR le quali, dato un documento, permettono di effettuarne l'analisi lessicale, estrarne i termini indici e memorizzare su disco in memoria permanente l'indice. Inoltre, grazie alle funzionalità fornite da queste primitive è possibile, partendo dall'esigenza informativa di un utente, procedere con la sua analisi

lessicale, ottenere i termini distinti dell'interrogazione e accedere all'indice al fine di reperire i documenti rilevanti all'esigenza informativa.

L'uso delle primitive di IR, quindi, ha avuto un ruolo importante nell'implementazione del modello; queste, infatti, hanno consentito di avere un accesso ottimizzato all'indice, portando di conseguenza ad un incremento nell'efficienza di 4IR.

3.3 Progetto del motore di ricerca

3.3.1 Indicizzazione

L'indicizzazione dei documenti della collezione è, come spiegato nella fase introduttiva, un passaggio fondamentale ai fini del funzionamento del motore di ricerca e del reperimento. Durante l'indicizzazione è possibile operare una rimozione delle *stop word* della collezione. Per *stop word* si intendono tutte quelle parole che, data la loro elevata frequenza in una lingua, sono ritenute poco significative in termine di potere discriminante in un documento; un esempio è dato dagli articoli e dalle preposizioni. Un'altra operazione possibile in fase di indicizzazione è lo *stemming* [1] ossia la riduzione delle parole originali alle radici; mediante lo *stemming* viene estratta, da ciascuna parola, una sotto-stringa, detta *stem*, che ne rappresenta la radice linguistica. Il vantaggio di lavorare con gli *stem* consiste nel fatto che risulta possibile far collassare varianti diverse della stessa radice riducendo di conseguenza la quantità di termini da indicizzare. Bisogna precisare che, mentre fino a pochi anni fa la rimozione delle *stop word* e l'applicazione dello *stemming* erano passaggi quasi obbligatori, perché permettevano di ridurre notevolmente la quantità di memoria destinata all'indice, oggi queste tecniche possono anche non essere eseguite in quanto, grazie alla disponibilità sempre crescente di memoria a basso costo, si preferisce indicizzare ogni singolo termine di un documento.

3.3.2 Trasformazione interrogazione-spettro

Un'operazione fondamentale nel modello LSPR è il passaggio dall'interrogazione allo spettro; questa fase vede l'interrogazione come la somma di più segnali sinusoidali, ognuno dato dal contributo di un termine dell'esigenza informativa. Verranno presentati ora i dettagli su come ottenere lo spettro di frequenze partendo da una data interrogazione.

[1] Nello sviluppo del sistema 4IR si è utilizzato lo *stemmer* di Porter.

3.3.2.1 Numero di campioni della DFT

Il primo passo da eseguire consiste nel ricavare i parametri necessari al calcolo della DFT: il numero di campioni N e il quanto di frequenza F. Si è deciso di impostare quest'ultimo ad un valore pari a 2 Hz e questa scelta è stata dettata dall'obiettivo di ottenere un effetto di dispersione spettrale; come spiegato nella sezione 2.1.1 si vuole infatti che la frequenza f_0 del segnale sinusoidale non sia un multiplo intero del quanto di frequenza F: in questo modo, avendo settato F pari a 2 Hz è sufficiente che le frequenze dei termini dell'interrogazione siano numeri dispari per ottenere la dispersione spettrale voluta.

Un'altra scelta che si è fatta è stata quella di stabilire che, per ogni termine dell'interrogazione, siano destinati 300 punti dello spettro e che il picco relativo al termine cada nel punto 200 di ogni intervallo. Il motivo di questa decisione sta nel fatto che, come dimostrato da prove sperimentali effettuate, allontanandosi di circa 100 punti sia a destra che a sinistra rispetto al punto dove è presente il picco, il modulo dello spettro della DFT vale meno dell'1% del valore di picco; in questo modo, in un intorno dei punti 100 e 300 di ogni intervallo, il modulo della DFT può considerarsi quasi esaurito.

Infine, si è deciso di inserire un intervallo di protezione di 300 punti, non associato a nessun termine dell'interrogazione, e posizionato subito dopo l'ultimo gruppo; così facendo ci si assicura che le componenti della DFT dell'ultimo termine dell'interrogazione siano praticamente nulle prima della metà dello spettro. Tenuto conto di quanto detto finora e considerato che, per il calcolo efficiente della DFT, N deve essere potenza di 2, la formula finale per il calcolo di N risulta essere:

$$N = 2^{\lceil \log_2(2 \cdot 300 \cdot (|Q|+1)) \rceil} \tag{3.1}$$

Dove $\lceil \rceil$ indica l'operazione di arrotondamento all'intero superiore. Per portare un esempio pratico di applicazione della formula 3.1 si consideri un'interrogazione composta da 2 termini; allora il calcolo di N risulta essere: $2^{\lceil \log_2(2 \cdot 300 \cdot (2+1)) \rceil} = 2^{\lceil \log_2(1800) \rceil} = 2^{11} = 2048$.

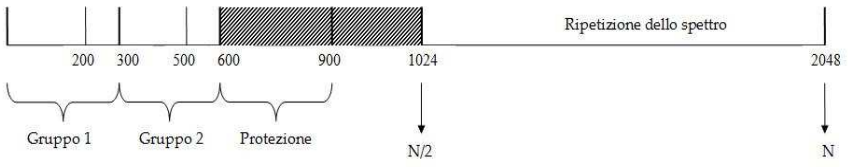


Figura 3.2: Suddivisione dei punti dello spettro di un'interrogazione formata da 2 termini

Per avere un'illustrazione riassuntiva, la figura 3.2 presenta la suddivisione dei punti dello spettro associato all'esigenza informativa secondo quanto enunciato precedentemente.

3.3.2.2 Frequenza associata ad ogni termine dell'interrogazione

Resta quindi da definire, una volta ottenuto il valore di N, come vengono calcolati i segnali sinusoidali che servono per ricavare i campioni sui quali verrà calcolata la DFT. Indicando con q_i ogni temine dell'interrogazione, la formula necessaria per ottenere il segnale sinusoidale associato a q_i è:

$$A_i \sin(2\pi f_i n T) \tag{3.2}$$

Nella 3.2 si è deciso di settare A_i pari al peso IDF del termine q_i dell'interrogazione nella collezione mentre f_i è stata scelta considerando che essa deve corrispondere circa al punto 200 di ogni gruppo e che per garantire l'effetto di dispersione spettrale, tale frequenza deve essere dispari. In definitiva, dunque, la formula che permette di calcolare f_i è:

$$f_i = (300 \cdot (i-1) + 200)F + 1, \qquad \text{i=1,2,...,|Q|} \tag{3.3}$$

3.3.2.3 Calcolo dello spettro

Dopo aver ottenuto le frequenze di tutti i termini dell'interrogazione bisogna valutare la somma dei |Q| segnali sinusoidali ottenuti su N punti per ottenere così il vettore su cui calcolare la DFT. Applicando quanto detto e chiamando *campioni* il vettore risultante si ha che:

$$campioni[n] = \sum_{i=1}^{|Q|} A_i \sin(2\pi f_i n T), \qquad \text{i=1,2,...,|N|} \tag{3.4}$$

Fatto ciò, non resta che applicare l'algoritmo per il calcolo della DFT al vettore *campioni*, ottenendo così lo spettro di frequenze associate all'interrogazione; questo algoritmo, dato un vettore in input, restituisce la DFT relativa a metà spettro in quanto, grazie alla simmetria, non è necessaria una valutazione di tutto lo spettro.

3.3.3 Trasformazione documento-filtro

Come spiegato nella metafora del modello LSPR in 2.2, una volta ottenuto lo spettro di frequenze associato all'interrogazione, i documenti della collezione sono visti come dei filtri che attenuano lo spettro. Resta da capire come vengono settati i parametri del filtro, il valore di ampiezza e i punti dove lo spettro deve essere completamente

attenuato. Per iniziare, si è assunto che i termini presenti in un documento con i loro relativi pesi siano i responsabili dei parametri del filtro che verrà creato. Un esempio di filtro è rappresentato in figura 3.3 in cui è rappresentata un'operazione di filtraggio nel primo gruppo di punti dello spettro (da 0 a 300) ma la stesso procedimento può essere esteso agli altri intervalli; qui si nota che due componenti dello spettro vengono completamente annullate mentre altre componenti vengono moltiplicate per un valore che va da 0 a 1.

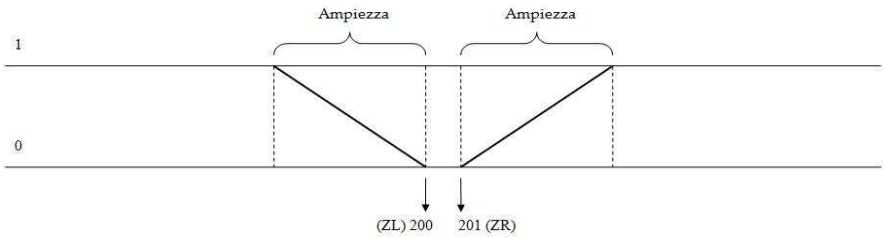


Figura 3.3: Operazione di filtraggio nel primo gruppo di punti (0-300)

Per scendere più nel dettaglio, si supponga di avere a disposizione un documento X_j e un'interrogazione Q della quale si è calcolato lo spettro; il documento in esame può produrre dei filtri per ogni termine dell'interrogazione che corrisponde ad un intervallo di 300 punti sullo spettro. Chiamando i l'i-esimo intervallo in esame che corrisponde quindi all'i-esimo termine delle interrogazioni, se nel documento X_j è presente tale termine, allora viene creato un filtro che azzera lo spettro in corrispondenza dei punti ZL(Zero Left) $= 300 * (i-1) + 200$ e ZR(Zero Right) $= 300 * (i-1) + 201$; non rimane quindi che determinare l'ampiezza del filtro, ottenuta moltiplicando una quantità fissa chiamata selettività (i cui dettagli saranno trattati in 4.3.2) per il peso del termine nel documento X_j e arrotondando il risultato all'intero più vicino. A questo punto si procede con l'operazione di filtraggio che consiste nel moltiplicare per dei valori compresi tra 0 e 1 i punti che si trovano negli intervalli: (ZL-ampiezza, ZL) e (ZR, ZR+ampiezza); per eseguire questa operazione, è sufficiente calcolare l'equazione della retta passante per i punti: (ZL-ampiezza, 1) e (ZL, 0) e l'equazione di quella passante per (ZR, 0) e (ZR+ampiezza, 1). Il valore per cui moltiplicare lo spettro sarà quindi dato dall'immagine del punto in esame sulla retta calcolata. Per chiarire quanto detto si ricorda che data una coppia di punti: (x_1,y_1) e (x_2,y_2) e chiamati m e q rispettivamente il coefficiente angolare e l'intercetta della retta, l'equazione della retta y passante per i due punti si ottiene nel seguente modo:

$$m = \frac{y_1 - y_2}{x_1 - x_2}$$
$$q = y_1 - m \cdot x_1$$

$$y = m \cdot x + q$$

Nel caso in cui nel documento non sia presente il termine dell'interrogazione, non viene creato nessun filtro.

3.3.4 *Ranking*

Dopo che si è trasformata l'interrogazione in spettro, filtrato lo spettro con ogni documento e ottenuto quindi, per ogni documento della collezione uno spettro filtrato, per calcolare la potenza di un documento si procede semplicemente sommando le componenti dello spettro associato. Secondo quanto ribadito nella metafora di LSPR, dati X_j e X_k due documenti delle collezione, X_j è da preferire a X_k se la potenza associata al primo è inferiore alla potenza associata al secondo; in base a questo principio, data un'interrogazione e dopo aver eseguito il filtraggio, i documenti della collezione vengono ordinati per potenza crescente; nella lista dei risultati ottenuti in risposta ad un'interrogazione, quindi, i documenti con potenza più bassa occuperanno i rank più elevati nella lista.

3.4 Implementazione del motore di ricerca

Lo sviluppo di 4IR si è concentrato sulla funzione di reperimento; per quanto riguarda la parte di indicizzazione, l'indice della collezione su cui poi sono stati fatti i test era già disponibile ed è indipendente dal modello di reperimento. Per questo motivo, in questo capitolo saranno presentati i dettagli relativi al reperimento dei documenti della collezione.

3.4.1 Metodi principali

L'implementazione di 4IR è stata fatta utilizzando il linguaggio di programmazione Java SDK versione 1.6, l 'ambiente di sviluppo NetBeans 7 e l'API Lucene versione 2.4.1. Il progetto consiste in una sola classe la quale contiene una serie di metodi, ognuno dedicato all'implementazione di una specifica funzionalità; i metodi Java principali che sono stati sviluppati sono:

4IR : Metodo principale che implementa il modello LSPR

analizza : Si occupa dell'analisi dell'esigenza informativa fornita in input

pesoIdfQuery : Calcola, data un'interrogazione, il valore di IDF di ogni suo termine

campioni : Restituisce il vettore su cui calcolare la DFT

calcolaDFT : Calcola la DFT del vettore fornito in ingresso

trasforma : Per ogni documento, esegue il filtraggio dell'interrogazione e calcola la potenza associata al documento dopo il filtraggio.

filtra : Implementa l'algoritmo di filtraggio

ranking : Ordina le potenze associate ai documenti in ordine crescente e restituisce la lista documento-potenza ordinata.

3.4.1.1 Analisi dell'interrogazione

La prima operazione che viene effettuata dall'algoritmo, data un'interrogazione, è procedere con la sua analisi lessicale. In questa prima fase, infatti, si riconoscono e rimuovono dall'interrogazione le *stop word*, in quanto sono portatrici di scarso contenuto informativo. Per indicare al sistema quali sono i termini da considerare *stop word* si può procedere o fornendo un file contenente tali termini oppure decidendo di utilizzare le *stop word* di default definite nella distribuzione di Lucene. Un'altra operazione che viene effettuata durante l'analisi dell'interrogazione è la riduzione di tutti i termini in minuscolo; quest'esigenza è dettata dal fatto che durante l'indicizzazione è stata effettuata la stessa operazione e che quindi l'indice contiene solo termini in minuscolo. Da qui si può intuire facilmente che se venissero confrontati due stessi termini, uno scritto in maiuscolo e un altro scritto in maiuscolo, il confronto darebbe esito negativo. Tutte queste operazioni di analisi lessicale sono state effettuate tramite l'ausilio di funzionalità fornite dalla libreria Lucene; una volta terminata l'analisi dell'esigenza informativa, si ottiene una nuova interrogazione i cui termini saranno impiegati per tutte le successive operazioni. D'ora in avanti, quando si farà riferimento all'interrogazione, si intenderà sempre quella ottenuta a seguito della fase di analisi.

3.4.1.2 Calcolo dello spettro associato all'interrogazione

Per arrivare ad ottenere lo spettro associato all'interrogazione, bisogna dapprima calcolare il valore di N e la frequenza f_i associata ad ogni termine dell'interrogazione. Queste operazioni vengono eseguite applicando le formule 3.1 e 3.3 presentate rispettivamente in 3.3.2.1 e 3.3.2.2. E' importante ricordare che, nel calcolo di N, bisogna eseguire un arrotondamento per eccesso del valore trovato; il motivo di tale operazione è da ricercare nell'inserimento di un intervallo di protezione dopo l'ultimo termine come spiegato in 3.3.2.1.

A questo punto è necessario procedere calcolando il peso IDF che ogni termine dell'interrogazione ha nella collezione. Per eseguire questa operazione sono state utilizzate delle primitive di Lucene che permettono di accedere all'indice e ottenere, oltre al numero totale di documenti, quelli che contengono un determinato termine; ricavati

questi valori, è sufficiente applicare la formula per il calcolo di IDF definita nella sezione 2.3. Le operazioni eseguite sono riassunte nello pseudocodice che segue.

Algoritmo: Calcola_pesi_IDF
Input: Query Q, indice della collezione I
Output: Vettore *pesiQuery* con i pesi di tutti i termini dell'interrogazione

1: accedi a I e ottieni il numero di documenti |C| della collezione
2: **for** $i = 1 \rightarrow |Q|$ **do**
3: accedi a I e ottieni il numero di documenti n_i che contengono il termine in esame
4: **if** n_i=0 **then**
5: $pesi\_Query[i] = 0$
6: **else**
$pesi\_Query[i] = \log_2\left(\frac{|C|+0.5}{n_i+0.5}\right)$
7: **end if**
8: **end for**
9: **return** $pesi\_Query$

Sono disponibili ora tutti gli ingredienti necessari per ottenere il vettore su cui sarà calcolata la DFT; basta infatti applicare la formula 3.4 e si ottiene un vettore di lunghezza N che contiene i valori del segnale associato all'interrogazione in ingresso. È sufficiente a questo punto fornire il vettore calcolato in ingresso all'algoritmo per il calcolo della DFT; tale algoritmo, essendo noto in letteratura, non viene riportato in questa tesi. Una volta ottenuta la DFT, essendo essa definita nel campo complesso, se ne calcola il modulo ottenendo così un vettore di numeri reali. Questi valori costituiscono quello che sarà lo spettro associato all'interrogazione in input. Per riassumere quanto detto, lo pseudocodice seguente riporta le operazioni che consentono, data un'interrogazione, di ottenere lo spettro ad essa associato; in questo codice DFT(x) rappresenta l'algoritmo che fornisce la DFT del vettore x in ingresso.

Algoritmo: Calcola_spettro
Input: Query Q, quanto di frequenza F, indice della collezione I
Output: Vettore *spettro* con i valori del modulo della DFT

1: $N = 2^{\lceil \log_2(2 \cdot 300 \cdot (|Q|+1)) \rceil}$
2: **for** $i = 1 \rightarrow |Q|$ **do**
3: $f_i = (300 \cdot (i-1) + 200)F + 1$
4: **end for**

5: $pesi\_query = Calcola\_pesi\_IDF(Q, I)$
6: **for** $n = 1 \rightarrow |N|$ **do**
7: $\quad campioni[n] = \sum_{i=1}^{|Q|} A_i \sin(2\pi f_i n T)$
8: **end for**
9: $spettro \leftarrow |DFT(campioni)|$
10: **return** *spettro*

3.4.1.3 Creazione dei filtri e filtraggio

Ora che si è ottenuto lo spettro associato all'interrogazione, si procede con l'operazione di filtraggio; come spiegato in 3.3.3 i documenti della collezione sono visti come dei filtri che attenuano lo spettro. Un controllo che è stato effettuato prima di procedere con la fase di filtraggio riguarda l'esistenza o meno di uno spettro da filtrare; questa operazione forse potrà risultare scontata ma è fondamentale: solo se lo spettro associato all'interrogazione non è nullo, allora si procede al filtraggio. Può succedere infatti che, data un'interrogazione, i suoi termini non compaiano in nessun documento della collezione. In questo caso il calcolo di N non incontrerebbe alcuna difficoltà, ma si otterrebbero pesi IDF nulli per ogni termine dell'esigenza informativa; di conseguenza, applicando la 3.4 si avrebbe un vettore *campioni* con tutte le componenti pari a 0 e anche lo spettro calcolato con la DFT sarebbe nullo. Proprio per questo motivo, non ha senso effettuare nessuna operazione di filtraggio e si conclude affermando che nessun documento è stato reperito per l'esigenza informativa fornita in ingresso. Posto quindi di aver inserito tale controllo, lo pseudocodice dei due algoritmi che seguono chiarisce rispettivamente come è stato implementato il passaggio da documento a filtro e come avviene l'operazione di filtraggio; si suppone inoltre di avere a disposizione un algoritmo Round(x) che arrotonda il numero reale x all'intero più vicino e una struttura *lista* che permetta di memorizzare l'identificativo di ogni documento con la potenza ad esso associata dopo il filtraggio.

L'algoritmo originale riportato in [Costa and Melucci, 2010] procedeva secondo un logica *Term-At-A-Time*, ossia si scorrevano tutti i termini dell'interrogazione e, per ognuno di essi, venivano reperiti i documenti della collezione in cui il termine appariva, procedendo con il relativo filtraggio. Il problema riscontrato in questa tecnica è che, quando un documento conteneva più termini dell'interrogazione, era necessario tener traccia, ad ogni termine, della potenza associata al documento per riuscire a filtrarlo nuovamente con i termini successivi. Nella versione implementata attualmente, invece, 4IR in prima battuta, data l'esigenza informativa, scorre tutti i termini di cui è composta e per ognuno di essi memorizza in una struttura dati i documenti che contengono tale termine. Successivamente, per effettuare il filtraggio, si procede secondo un approccio *Document-At-A-Time* in cui cioè vengono scorsi i documenti memorizzati

e per ognuno si filtra lo spettro associato in base ai termini dell'interrogazione presenti nel documento. Questo permette innanzitutto di avere una sostanziale riduzione dei documenti che devono essere filtrati; infatti saranno processati solo quelli in cui è contenuto almeno un termine dell'esigenza informativa. Inoltre, secondo questo modo di procedere, inizialmente si ha un picco di risorse computazionali usate per accedere all'indice e reperire i documenti; dopo questa fase, però, l'utilizzo di memoria dovuto allo scorrimento dei documenti memorizzati nella struttura dati è minore e costante .

Algoritmo: Trasforma
Input: Query Q, quanto di frequenza F, indice della collezione I
Output: Struttura *lista* con gli identificativi di documento e potenza associata.

1: $spettro = Calcola\_spettro(Q, F, I)$
2: crea una struttura dinamica *documenti*
3: **for** $i = 1 \rightarrow |Q|$ **do**
4: $ZL[i] = 300 \cdot (i - 1) + 200$
5: ottengo la posting list P di q_i
6: **while** ci sono ancora documenti in P **do**
7: ottieni l'identificativo d del documento corrente
8: **if** d non è già presente in *documenti* **then**
9: aggingi d a *documenti*
10: **end if**
11: **end while**
12: **end for**
13: crea *lista* di dimensione uguale a *documenti*
14: **for** $i = 1 \rightarrow |documenti|$ **do**
15: $spettro\_filtrato = spettro$
16: **for** $j = 1 \rightarrow |Q|$ **do**
17: **if** $q_j \in documenti[i]$ **then**
18: $ampiezza = Round$(selettività ·peso di q_j in $documenti[i]$)
19: $spettro\_filtrato$ =Filtra($spettro\_filtrato, ZL[j], ampiezza$)
20: **end if**
21: **end for**
22: $potenza[i] = \sum_{k=1}^{|spettro\_filtrato|} spettro\_filtrato[k]$
23: aggiungi $documenti[i]$ e $potenza[i]$ a $lista$
24: **end for**
25: **return** *lista*

Algoritmo: Filtra

Input: Spettro S, valore di ZL, ampiezza A

Output: Spettro della query filtrato $spettro\_filtrato$

1: $spettro\_filtrato$=$spettro$
2: ZR=ZL+1
3: soglia_sinistra = ZL - ampiezza
4: soglia_destra = ZR + ampiezza
5: $spettro\_filtrato$[ZL] = 0
6: $spettro\_filtrato$[ZR] = 0
7: // Calcolo della retta passante per i punti (soglia_sinistra, 1) e (ZL, 0)
8: m =(1 - 0) / (soglia_sinistra - ZL)
9: q = 1 - m · soglia_sinistra
10: **for** $x = soglia\_sinistra \rightarrow ZL$ **do**
11: y = m · x + q
12: $spettro\_filtrato$[x] = $spettro\_filtrato$[x] · y
13: **end for**
14: // Calcolo della retta passante per i punti (ZR, 0) e (soglia_destra, 1)
15: m =(1 - 0) / (soglia_destra - ZR)
16: q = 1 - m · soglia_destra
17: **for** $x = ZR \rightarrow soglia\_destra$ **do**
18: y = m · x + q
19: $spettro\_filtrato$[x] = $spettro\_filtrato$[x] · y
20: **end for**
21: **return** $spettro\_filtrato$

3.4.2 Funzionamento di 4IR

Posto di avere a disposizione una collezione sperimentale ed di aver effettuato l'operazione di indicizzazione, l'algoritmo prevede che in ingresso sia fornita una stringa rappresentante l'esigenza informativa da soddisfare. Nell'implementazione attuale, 4IR è stato progettato per leggere le interrogazioni da un file di testo e procedere con il reperimento; un parametro da fornire in ingresso riguarda quindi il percorso in cui trovare il file di testo ed estrarne le informazioni. Un'altra indicazione che è necessario fornire affinché il motore possa funzionare è il percorso dove si trova l'indice della collezione; senza questo dato, infatti, non sarebbe possibile reperire alcun documento. Non rimane quindi che specificare percorso e nome del file di output dove verranno stampati i risultati del reperimento.

Settate queste variabili, 4IR procede con il calcolo del numero di campioni N, della frequenza f_i associata ad ogni termine dell'interrogazione e del loro peso IDF. A questo punto si ottiene il vettore su cui calcolare la DFT e si procede al calcolo di quest'ultima; infine, ottenuto il modulo della trasformata discreta di Fourier si hanno a disposizione tutti i valori dello spettro che, se sommati, portano ad avere la potenza associata all'interrogazione di input. Si procede quindi con la fase di filtraggio e quella di *ranking* ottenendo così, in risposta all'esigenza informativa, la lista ordinata per rilevanza dei documenti con relativa potenza associata. I dati, vengono formattati nel formato richiesto dal software Trec_eval [2] e memorizzati nel file di output indicato dall'utente.

[2]Il software sarà presentato nella sezione 4.1

CAPITOLO

QUATTRO

ESPERIMENTI

Posto ora di aver implementato il modello LSPR come descritto nel capitolo 3, risulta di notevole interesse avere un'indicazione sulle *performance* e utilizzare tali indicazioni per confrontare 4IR con un altro o con un'altra versione di se stesso: si deve quindi eseguire la valutazione del modello di IR. In questa fase vengono utilizzate alcune misure, descritte in dettaglio nella sezione 4.1, che servono per avere degli indici di riferimento, relativi alle caratteristiche del modello che si vogliono testare. Un altro elemento importante che entra in gioco nella fase di test è la collezione sperimentale, trattata nella sezione 4.2 che fornisce l'insieme di documenti, interrogazioni e giudizi di rilevanza sui documenti della collezione da utilizzare per la valutazione del sistema. In questa tesi il confronto è stato effettuato tra il modello LSPR sviluppato e il modello probabilistico BM25 considerato lo stato dell'arte in letteratura. I test sono quindi stati svolti per verificare l'ipotesi che il modello LSPR implementato da 4IR dimostrasse prestazioni confrontabili con quelle di BM25; se questo fosse stato confermato, infatti, si sarebbe potuto pensare di investire maggiormente in LSPR per arrivare ad utilizzarlo in contesti come i motori di ricerca per il WWW o nelle reti P2P.

4.1 Misure

E' necessario stabilire alcune misure da utilizzare come indicatori che riflettano l'efficacia e l'efficienza di un sistema di IR; queste misure devono essere necessariamente di tipo quantitativo per poter operare un confronto. Si ricorda che per efficacia si intende la quantità di documenti rilevanti che sono stati reperiti mentre con efficienza ci si riferisce alle risorse computazionali utilizzate e alle funzionalità dell'interfaccia tra l'utente e il sistema. Richiamo e precisione sono le misure d'efficacia di riferimento; chiamando P la precisione, R il richiamo, A il numero di documenti rilevanti ad un'esi-

genza informativa e B il numero di documenti reperiti in risposta ad un'interrogazione, possiamo definire precisione e richiamo come segue:

$$P = \frac{|A \cap B|}{|B|} \qquad R = \frac{|A \cap B|}{|A|} \tag{4.1}$$

Dalla formula 4.1 si evince che il richiamo si riferisce al numero di documenti rilevanti reperiti tra quelli rilevanti ad un'interrogazione mentre la precisione riguarda la proporzione di documenti rilevanti tra quelli reperiti. Un'altra cosa che è possibile constatare è la relazione inversa che lega P e R; infatti per aumentare il richiamo è necessario reperire una maggiore quantità di documenti e, di conseguenza, ridurre la precisione a meno che ogni documento in più reperito sia rilevante. L'utilizzo dei soli indicatori di richiamo e precisione per valutare un sistema di IR può portare a dei problemi di calcolo; per citare degli esempi, P non è calcolabile qualora nessun documento venga reperito dal sistema così come R non è calcolabile se non è noto il numero di documenti rilevanti ad una certa interrogazione.

A

Rank	1	2	3	4	5	6
Precisione	0	1/2	2/3	2/4	2/5	3/6

B

Rank	1	2	3	4	5	6
Precisione	1/1	2/2	2/3	2/4	3/5	4/6

Figura 4.1: Calcolo della precisione per due diverse liste di risultati

Una delle misure di efficacia utilizzate durante i test per valutare il motore è il MAP (*Mean Average Precision*); prima di definirla è necessario introdurre un'altra misura, AP (*Average Precision*). Quest'ultima è la media dei valori di precisione calcolati ad ogni documento rilevante tra quelli restituiti in risposta ad un'interrogazione; MAP è quindi la media delle precisioni medie non interpolate. Per meglio chiarire come viene calcolato il MAP, si prenda in considerazione la figura 4.1 che riporta i primi 6 documenti reperiti in risposta ad un'interrogazione per due diverse configurazioni di un sistema e con il valore di precisione calcolato ad ogni documento; in questa illustrazione i documenti evidenziati in neretto sono quelli rilevanti. Nell'esempio di figura 4.1 il calcolo di AP per le due configurazioni risulta essere rispettivamente:

$$\frac{1}{3} \cdot \left(\frac{1}{2} + \frac{2}{3} + \frac{1}{2}\right) = \frac{5}{9} \qquad \frac{1}{4} \cdot \left(1 + 1 + \frac{3}{5} + \frac{2}{3}\right) = \frac{49}{60}$$

mentre il MAP risulta:

$$\frac{1}{2} \cdot \left(\frac{5}{9} + \frac{49}{60}\right) = \frac{247}{360}$$

Altri indici di efficacia che si sono affermati nell'ambito della valutazione di un sistema di IR sono il CG (*Cumulative Gain*), ossia il guadagno cumulato medio sul numero di documenti trovati. L'idea alla base di CG sta nel fatto che, scorrendo una lista di risultati ordinata per rilevanza decrescente, ogni volta che si incontra un documento rilevante si accumula un guadagno pari al grado di rilevanza del documento stesso; se al posto della posizione in cui si trova un documento se ne considera il logaritmo in base 2, si ottiene il *Discounted Cumulative Gain* (DCG). Quanto detto è formalizzato nelle due espressioni seguenti dove i indica la posizione nella lista e r_i è il grado di rilevanza del documento in posizione i:

$$CG = \frac{r_i}{i} \qquad DCG = \frac{r_i}{\log_2 i} \tag{4.2}$$

Se i documenti reperiti in risposta ad un'interrogazione vengono ordinati per grado di rilevanza decrescente, si ottiene il valore di DCG ideale, che può essere utilizzato per normalizzare DCG e ottenere la misura chiamata NDCG (*Normalized Discounted Cumulative Gain*):

$$NDCG = \frac{DCG}{DCG^*} \qquad DCG^* = max(DCG) \tag{4.3}$$

Durante i test di valutazione dei modelli, per valutarne l'efficacia, si è preso in considerazione: MAP, NDGC, la precisione e il NDCG dopo i primi 10 documenti reperiti; questo è stato fatto per avere un'indicazione sulla capacità del modello di posizionare ai primi posti i documenti maggiormente rilevanti.

Trec_Eval

Per poter effettuare la valutazione di diversi sistemi di IR e far in modo che le misure calcolate fossero le stesse per tutti i motori testati, è stato introdotto uno strumento software chiamato Trec_eval. Era necessario, infatti, poter avere una visione standardizzata dei risultati ottenuti da un motore di ricerca su una collezione sperimentale; per raggiungere questo obiettivo bisognava sviluppare un programma che consentisse di analizzare i risultati ottenuti dai vari sistemi di IR in modo da poterli confrontare e avere un'indicazione su quale, tra quelli in esame, avesse le migliori prestazioni. Trec_eval nasce proprio per far fronte a questa esigenza ed è lo strumento che viene utilizzato per valutare un sistema di IR. [1]

Il software, per funzionare, ha bisogno di ricevere in input la lista dei primi k documenti (dove k viene scelto da chi effettua la valutazione) reperiti dal motore di ricerca per ciascuno dei *topic* presenti nella collezione di test e un file contenente i giudizi di rilevanza per i *topic* considerati. Il file relativo ai giudizi di rilevanza è

[1] Il software è liberamente scaricabile all'URL **http://trec.nist.gov/trec_eval**

solitamente chiamato qrels.txt ed è disponibile all'indirizzo web di TREC [2] mentre il file contenente la lista dei primi k documenti reperiti ad ogni *topic* deve essere prodotto dallo sperimentatore e dovrà avere necessariamente la seguente formattazione:

n_topic *O* *Id_doc* *rank* *sim* *STANDARD*

dove i vari campi vanno così interpretati:

- *n_topic*: numero del topic in esame
- *O*: campo costante, non deve essere modificato
- *Id_doc*: identificativo TREC del documento reperito dal sistema in risposta al topic n_topic
- *rank*: posizione che occupa il documento Id_doc nella lista di documenti in risposta al topic n_topic
- *sim*: valore (float) di similarità fra documento e topic; sim è supposta essere maggiore per i documenti reperiti prima
- *STANDARD*: campo costante, non deve essere modificato

Una volta che si ha a disposizione il file con i giudizi di rilevanza per ogni *topic* e che il file con i risultati della valutazione del modello, appositamente formattato, è stato prodotto, è possibile procedere con la fase di valutazione del motore di ricerca. Un esempio di esecuzione del programma Trec_eval è visualizzabile in figura 4.2; per maggiori dettagli sul funzionamento di Trec_eval e sulle varie misure che esso è in grado di fornire si rimanda a [Grossman, 2005]. Durante i test, come già spiegato nella sezione 4.1 le misure di efficacia che sono state considerate sono MAP, NDGC, precisione e NDCG dopo i primi 10 documenti reperiti.

Figura 4.2: Esempio di utilizzo del software trec_eval

[2] Il file qrels.txt per la collezione "trec 2001 web trac ad hoc topics" usata nei test è disponibile all'URL **http://trec.nist.gov/data/t10.web.html**

4.2 Collezione sperimentale

Uno dei requisiti di base della fase di valutazione è che i risultati ottenuti con diverse configurazioni possano essere confrontati tra loro. Per permettere questo confronto e assicurare che gli esperimenti fatti siano ripetibili è necessario che sia la collezione che le interrogazioni siano le stesse per tutte le configurazioni testate. Inoltre, dato che il calcolo di tutte le misure di efficacia prevede la conoscenza del numero di documenti rilevanti, se questi sono ignoti non è possibile avere indicazioni sulle performance del motore. Per affrontare questi problemi è stato introdotto il concetto di collezione sperimentale (*test collection*): un insieme di documenti, interrogazioni e giudizi di rilevanza espressi per alcuni documenti reperiti in risposta ad un'esigenza informativa; in questo modo, avendo a disposizione i giudizi di rilevanza per i documenti reperiti a fronte di un'interrogazione, e di conseguenza anche il numero di documenti rilevanti, il calcolo delle misure di efficacia può essere fatto senza problemi.

L'utilizzo di collezioni sperimentali è una pratica comune in IR e nel corso degli anni sono state create diverse collezioni di test che si differenziano soprattutto per la quantità di documenti e interrogazioni presenti e per l'argomento trattato nei documenti della collezione. La fase di valutazione consiste in pratica, avendo una collezione sperimentale e un motore di ricerca, nel sottoporre un'esigenza informativa al sistema sviluppato e ottenere quindi una lista di documenti reperiti in risposta all'interrogazione; a questo punto si andranno a confrontare i documenti rilevanti e i rispettivi giudizi di rilevanza con i documenti reperiti dal motore. Più un motore riesce a reperire e posizionare nei primi posti i documenti rilevanti all'esigenza informativa, più il motore è performante.

A livello internazionale, sono diverse le iniziative di valutazione che vengono condotte; fra queste vale la pena citare TREC (*Text REtrieval Conference*), una serie di *workshop* organizzati dal *National Institute of TEchnology* (NIST)[3] con lo scopo di definire un modello da utilizzare per la valutazione su grandi moli di documenti, soprattutto di tipo testuale, dei metodi dell'IR. Nel corso degli anni, le collezioni di test prodotte sono cambiate per riflettere i cambiamenti nel comportamento e nelle esigenze di ricerca degli utenti. Per ogni collezione sperimentale prodotta nell'ambito di TREC, sono disponibili delle esigenze informative da soddisfare denominate *topic* contenenti tre campi principali delimitati da appositi tag. Un esempio di un *topic* TREC è visibile in figura 4.3 dove si nota la presenza del campo *title*, supposto essere un'interrogazione di lunghezza ridotta, che rispecchia la tipica esigenza informativa di un utente che opera nel web. Il campo *description*, invece, non è altro che una versione più lunga dell'interrogazione contenuta in *title*, fornendo in questo modo un dettaglio

[3] http://trec.nist.gov/

maggiore sull'esigenza informativa. Infine, troviamo il campo *narrative* che descrive i criteri e i requisiti che un documento deve soddisfare affiché possa essere ritenuto rilevante all'esigenza informativa in questione; queste informazioni sono utilizzate da chi dovrà stabilire quali sono e con che grado i documenti rilevanti all'interrogazione.

Collezione	Dimensione	Numero di documenti	Numero di interrogazioni
CACM	2.2 Mb	3204	64
trec 2001 web trac ad hoc topics	10 Gb	1692096	50

Tabella 4.1: Alcune collezioni sperimentali sviluppate nell'ambito TREC

<top>

<num> Number: 501
<title> deduction and induction in English?

<desc> Description:
What is the difference between deduction and induction in the process of reasoning?

<narr> Narrative:
A relevant document will contrast inductive and deductive reasoning. A document that discusses only one or the other is not relevant.

</top>

Figura 4.3: Esempio di un topic TREC

Durante la fase di test di 4IR è stata utilizzata la collezione sperimentale "*TREC 2001 web trac ad hoc topics*" che fa parte della collezione di test WT10g e conta 1.692.096 documenti e 50 interrogazioni; nella tabella 4.1 sono riportati maggiori dettagli sulla collezione usata per i test ed è presentato un raffronto con un'altra collezione sperimentale, denominata CACM contenente 3.204 documenti e utilizzata in [Costa, 2009] per i test su LSPR. Si è deciso di utilizzare una collezione con quasi 1,7 milioni di documenti proprio perchè, come spiegato in 1.2, lo scopo dei test era quello di verificare se l'efficacia del modello LSPR riscontrata durante la valutazione su un piccolo insieme di documenti, fosse confermata anche quando il modello operava in una collezione più vasta.

4.3 Metodologia sperimentale

Quello che si è cercato di fare nella fase di test è stato un confronto fra le prestazioni del modello LSPR e quelle di BM25. Oltre ad un semplice raffronto numerico relativo agli indici di efficacia dei due modelli, è stato fatto anche un test statistico di significatività per vedere se le differenze riscontrate durante la valutazione fossero significative da un punto di vista statistico o dovessero invece essere considerate solo frutto della naturale variabilità dei dati.

In sostanza, scelta una misura di efficacia, ad esempio AP, e ottenuti i valori di AP per tutti i 50 *topic* della collezione sperimentale e per entrambi i modelli testati, è stato effettuato un test di Student per dati appaiati. L'utilizzo di questo test è stato dettato dal fatto che, disponendo di misure ripetute sulle stesse unità statistiche (i *topic*), esiste un'associazione tra i valori ottenuti con BM25 e quelli ottenuti con LSPR. In figura 4.4 è riportato il diagramma di dispersione dei valori di AP per LSPR e BM25: l'associazione fra i due modelli appare evidente. Non è possibile

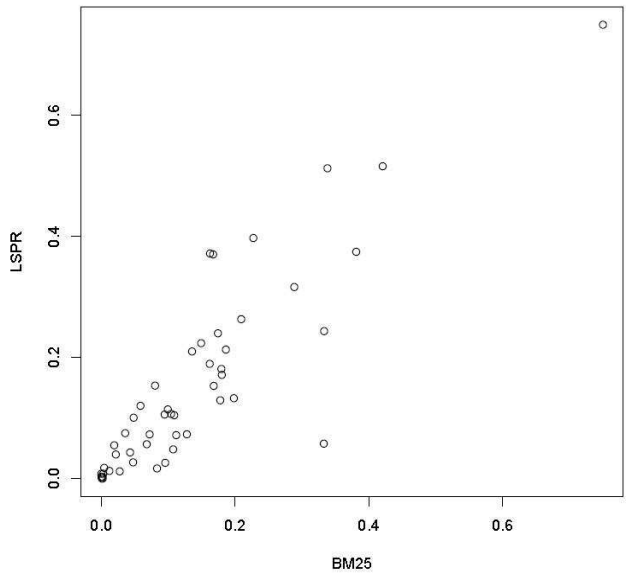


Figura 4.4: Diagramma a dispersione dei valori di AP per LSPR e BM25

quindi utilizzare un test di Student a due campioni ma è necessario applicare il test di Student per dati appaiati che contempla la presenza di un'effetto del *topic* sui valori di AP ottenuti. I dettagli statistici sull'analisi di significatività sono riportati nella sezione 4.4.3, dove verrà dimostrato che il problema di verificare la differenza di AP nei due campioni sarà affrontato utilizzando un test di Student ad un campione. E' importante

ricordare che un'assunzione su cui si basa il test vede la normalità dei dati del campione analizzato; è possibile dimostrare tuttavia che il test e le procedure rimangono valide anche se la distribuzione all'interno dei due gruppi non è normale purché la numerosità campionaria sia "sufficientemente grande". Una semplice regola per stabilire quando una numerosità campionaria può essere ritenuta "sufficientemente grande" suggerisce che i dati presenti nel campione devono essere in numero maggiore o uguale a trenta se la distribuzione dei dati è, almeno approssimativamente, simmetrica. Esistono tuttavia casi in cui, nonostante l'abbondanza delle unità statistiche a disposizione, l'assunzione di normalità per i dati è piuttosto forzata. Nel caso in esame, prima di effettuare il test di Student ad un campione si è effettuata un'analisi esplorativa dei dati indagando anche sull'ipotesi di normalità mediante un procedimento grafico (*normal probability plot*) e uno analitico (test di normalità di Shapiro-Wilk). Tutti i risultati e i dettagli procedurali sono presentati nella sezione 4.4.3. Inoltre, per potersi svincolare il più possibile da ipotesi sulla distribuzione dei dati, è stato effettuato anche l'equivalente del test di Student per dati appaiati nell'ambito non parametrico, il test di Wilcoxon.

Durante la fase di valutazione di LSPR si è proceduto in due direzioni: dapprima è stato effettuato un test considerando come esigenza informativa il testo riportato nel campo *title* di ogni *topic*, poi lo stesso test è stato fatto prendendo in considerazione il contenuto di *description*. Il motivo di questa scelta è che si voleva avere un'idea di come variassero le prestazioni dei modelli BM25 e LSPR passando da un'interrogazione con un numero ristretto di termini (*title*) ad un'altra più lunga e dettagliata (*description*).

Un'altra prova fatta ha visto l'utilizzo di diversi insiemi di termini da considerare *stop word*, e quindi da rimuovere, nella fase di analisi dell'interrogazione. Sia per il campo *title* che per *description*, infatti, è stata fatta una prima valutazione di BM25 e LSPR considerando come "rumore" un insieme ristretto di termini predefiniti in Lucene; in seconda battuta, invece, è stato ripetuto lo stesso test impostando come *stop word* tutti i termini portatori di scarso contenuto informativo, raccolti in un *file*. Il motivo di questa prova è che si voleva analizzare, nei modelli considerati, come variavano le misure di efficacia inserendo o meno alcune *stop word* nell'esigenza informativa..

Nel seguito verranno presentati i dettagli relativi al settaggio dei parametri di BM25 e LSPR.

4.3.1 Baseline BM25

Partendo dalla formula mediante la quale si ottiene il peso di un termine in un documento nel modello BM25:

$$w_i^{BM25} = \frac{TF}{k_1\left((1-b)+b\frac{dl}{avdl}\right)+TF} \cdot \log\left(\frac{(r_i+0.5)(|C|-R+r_i+0.5)}{(n_i-r_i+0.5)(R-r_i+0.5)}\right) \qquad (4.4)$$

i valori di TF, N, n_i, dl, $avdl$ risultano noti mentre rimangono da settare i parametri R,r_i,b e k_1. Come già spigato in 2.3, qualora, come in questo caso, non sia disponibile alcun indizio sulla rilevanza dei documenti, è sufficiente settare il valore di R e r_i a 0. Resta quindi da definire il valore delle costanti b e k_1; ricordando i vincoli $0.5 \leq b \leq 0.8$ e $1.2 \leq k_1 \leq 2$ si è scelto di porre $b = 0.8$ e $k_1 = 2$. La formula finale BM25 per calcolare il peso di un termine nel documento risulta quindi essere:

$$w_i^{BM25} = \frac{TF}{2\left(0.2 + 0.8\frac{dl}{avdl}\right) + TF} \cdot \log\left(\frac{((|C| + 0.5)}{(n_i + 0.5)}\right) \tag{4.5}$$

4.3.2 LSPR

Per quanto riguarda l'algoritmo LSPR, è necessario fornire indicazioni sulla scelta di alcuni parametri del modello. Innanzitutto, nella formula 3.3 si è impostato il valore della frequenza F pari a 2; il motivo di questa scelta è che si vuole garantire l'effetto di dispersione spettrale. Un'altra scelta che è stata fatta, nella 3.2, è stata quella di porre A_i pari al peso IDF del temine i nell'interrogazione; questo è stato fatto perché, da prove effettuate, con l'utilizzo dello schema di pesatura IDF si è ottenuta l'efficacia migliore. Infine, nella fase di filtraggio, l'ampiezza del filtro è ottenuta moltiplicando una quantità fissa chiamata selettività per il peso del termine nel documento e arrotondando il risultato all'intero più vicino. Si è deciso di impostare il valore di selettività pari a 40; questa scelta è stata fatta perché si è visto che per valori maggiori o minori il MAP diminuisce, come se questo rappresentasse un punto di massimo di una qualche funzione che lega MAP a selettività. Il peso del termine nel documento è stato invece calcolato utilizzando il sistema di pesatura BM25, applicando la formula 4.5.

4.4 Risultati sperimentali

Vengono ora presentati i risultati ottenuti dalla valutazione del modello LSPR e dal suo confronto con la *baseline* BM25. Una precisazione da fare riguarda il fatto che i risultati esposti sono stati ottenuti rimuovendo dall'interrogazione tutte le *stop word* più comuni. Infatti, da prove effettuate, si è visto che quando nell'esigenza informativa sono presenti alcuni termini "rumore" l'efficacia del modello diminuisce; da qui, la necessità di rimuovere tutte le *stop word* dall'interrogazione nella fase di analisi. Bisogna evidenziare però, come tale scelta abbia portato ad alcuni problemi in un *topic*, per la precisione il numero 531; esso, presentando nel campo *title* una stringa formata esclusivamente da *stop word*, porta l'algoritmo LSPR ad eliminare tutti i termini dell'interrogazione durante la fase di analisi. Solo nel caso del test su *title*, quindi, il *topic* 531 non è stato considerato per calcolo degli indici di efficenza di LSPR e BM25

in quanto non apporta nessun contributo informativo. Fra le varie misure considerate, inoltre, si riportano qui solo i risultati relativi al MAP in quanto è l'indice che si è maggiormente imposto negli ultimi anni e permette di avere una visione globale dell'efficacia di un modello, anche se saranno fatti dei brevi commenti sulle altre misure di efficacia calcolate. I risultati presentati sono stati ottenuti con l'utilizzo del software trec_eval descritto in 4.1 e impostando gli algoritmi LSPR e BM25 in modo che restituissero, per ogni *topic*, la lista dei primi 1000 documenti reperiti.

4.4.1 Test su *title*

Considerando come interrogazione la stringa riportata nel campo *title* di ogni *topic* della collezione di test, si è ottenuto il valore del MAP per il modello BM25 e LSPR. I risultati ottenuti sono riassunti in figura 4.5. Come si può notare, il modello LSPR

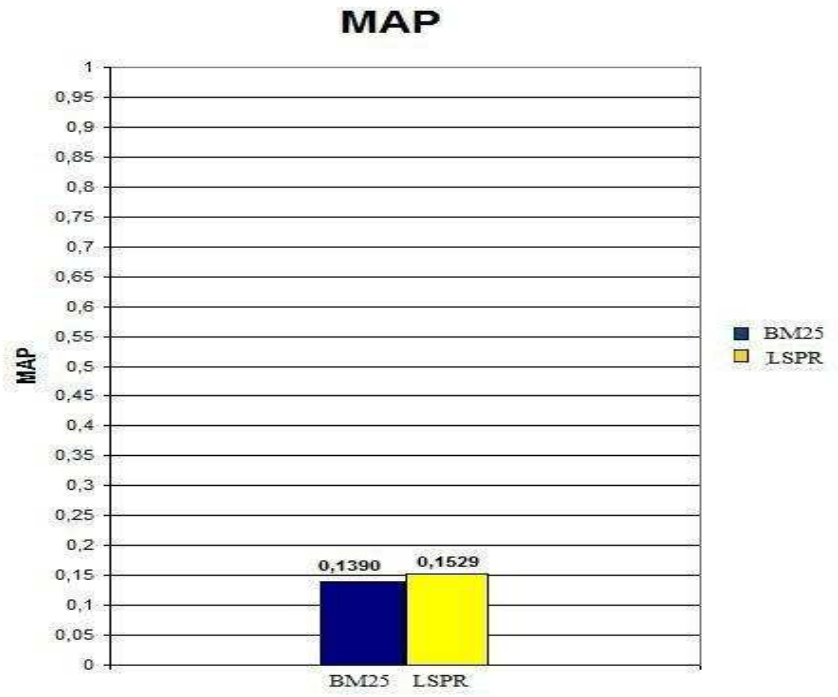


Figura 4.5: Valori del MAP per LSPR e BM25 utilizzando come interrogazione il campo *title*

presenta un MAP superiore rispetto a quello di BM25 di riferimento; nella sezione 4.4.3 sarà effettuato il test statistico t di Student per dati appaiati per avere un'indicazione sul fatto che la differenza riscontrata sia o meno significativa da un punto di vista statistico. Resta comunque il fatto che, considerando l'indice MAP, le prestazioni

di LSPR appaiono confrontabili con quelle della *baseline* BM25. Per completezza, in figura 4.6 sono riportati i valori di AP ottenuti da entrambi i modelli in ogni *topic* della collezione sperimentale; come si può facilmente verificare, il valore del MAP è ottenuto facendo la media aritmetica di tutti i valori di AP. Per quanto riguarda le altre misure

TOPIC	BM25	LSPR
501	0,0196	0,0549
502	0,0958	0,026
503	0,0486	0,1005
504	0,2887	0,3165
505	0,1627	0,372
506	0,1077	0,0482
507	0,0588	0,1202
508	0,022	0,0397
509	0,2094	0,2633
510	0,3383	0,5128
511	0,1359	0,21
512	0,1804	0,1713
513	0,1046	0,1067
514	0,1683	0,1531
515	0,0806	0,1536
516	0,0433	0,0431
517	0,0682	0,0567
518	0,0278	0,0118
519	0,1093	0,1046
520	0,0478	0,0267
521	0,0024	0,0001
522	0,036	0,0749
523	0,0836	0,0167
524	0,002	0,0027
525	0,1797	0,1811

TOPIC	BM25	LSPR
526	0,0725	0,0729
527	0,2275	0,3975
528	0,3812	0,3747
529	0,1623	0,1895
530	0,1674	0,3704
532	0,1865	0,2131
533	0,1746	0,24
534	0,0006	0,003
535	0,0032	0,008
536	0,1495	0,2237
537	0,0046	0,0176
538	0,3333	0,2436
539	0,1282	0,0731
540	0,0949	0,1058
541	0,1984	0,1327
542	0,0009	0,0002
543	0,0003	0,0079
544	0,4208	0,5161
545	0,3328	0,0576
546	0,0124	0,0126
547	0,1122	0,0717
548	0,75	0,75
549	0,178	0,1293
550	0,0997	0,1142

Figura 4.6: Valori di AP ottenuti da LSPR e BM25 in ogni *topic*

di efficacia che si sono considerate, ossia NDCG, precisione e NDCG dopo i primi 10 documenti reperiti, la situazione che emerge non è così chiara; infatti, in alcuni *topic* il modello LSPR sembra avere prestazioni superiori di BM25, in altri invece la situazione appare ribaltata.

4.4.2 Test su *description*

Assumendo ora che l'esigenza informativa da soddisfare sia rappresentata dal contenuto del campo *description* di ogni *topic*, come nel caso precedente con l'utilizzo di trec_eval si sono ottenuti i valori del MAP per LSPR e BM25 e la situazione è stata riassunta in figura 4.7. Come nel caso precedente, si riportano in figura 4.8 i valori di AP ottenuti per ogni *topic* della collezione di test.

Dai risultati sul MAP emerge che il modello BM25 sembra leggermente superiore rispetto a LSPR anche se la differenza riscontrata è dell'ordine dei millesimi e pratica-

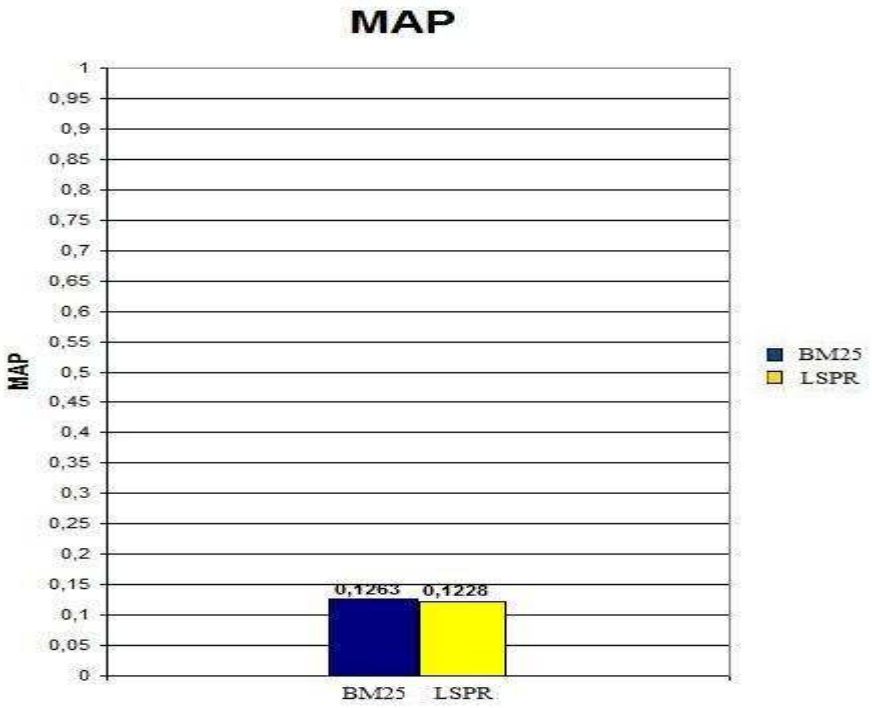


Figura 4.7: Valori del MAP per LSPR e BM25 utilizzando come interrogazione il campo *description*

TOPIC	BM25	LSPR
501	0,02	0,08
502	0,10	0,03
503	0,02	0,06
504	0,18	0,13
505	0,18	0,37
506	0,11	0,05
507	0,03	0,14
508	0,03	0,04
509	0,17	0,27
510	0,29	0,26
511	0,11	0,12
512	0,18	0,17
513	0,30	0,32
514	0,10	0,04
515	0,10	0,18
516	0,04	0,04
517	0,04	0,04
518	0,04	0,02
519	0,10	0,14
520	0,08	0,14
521	0,01	0,00
522	0,08	0,31
523	0,10	0,03
524	0,03	0,04
525	0,22	0,09

TOPIC	BM25	LSPR
526	0,05	0,07
527	0,12	0,17
528	0,38	0,37
529	0,18	0,15
530	0,05	0,09
531	0,00	0,01
532	0,24	0,22
533	0,14	0,18
534	0,00	0,00
535	0,00	0,00
536	0,07	0,15
537	0,00	0,00
538	0,29	0,04
539	0,13	0,07
540	0,03	0,07
541	0,20	0,13
542	0,27	0,24
543	0,00	0,00
544	0,24	0,13
545	0,42	0,21
546	0,11	0,14
547	0,11	0,07
548	0,31	0,23
549	0,26	0,22
550	0,08	0,08

Figura 4.8: Valori di AP ottenuti da LSPR e BM25 in ogni *topic*

mente trascurabile; i due modelli confrontati sembrano quindi equivalersi per quanto riguarda il MAP quando viene utilizzato come interrogazione il contenuto del campo *description*. Un'altra cosa che si evidenzia dal confronto fra il grafico in figura 4.5 e quello in figura 4.7 è che, quando l'interrogazione è più lunga, il MAP di entrambi i modelli, in particolare di LSPR, si abbassa; da questa considerazione sembra quindi preferibile utilizzare il modello LSPR quando l'esigenza informativa è composta da un numero ridotto di termini.

4.4.3 Analisi di significatività

Verranno ora presentati i risultati del test t di Student per dati appaiati che è stato usato per indagare sulla differenza in termini di MAP riscontrata nei test; in questo caso si pone l'attenzione sui risultati ottenuti considerando, come interrogazione, la stringa contenuta nel campo *title*. I due campioni analizzati sono dati dai valori di AP per ogni *topic* (ad eccezione del numero 531 per quanto spiegato in 4.4) ottenuti rispettivamente con il modello LSPR e BM25; la numerosità campionaria, quindi, è pari a 49 unità in entrambi i campioni.

Partendo dai dati campionari, si procede calcolando la differenza fra i valori di AP ottenuti con LSPR e quelli ottenuti con BM25; in questo modo si ottiene un campione di differenze z di numerosità 49. Il problema di verificare se i valori di AP nei due campioni possano essere ritenuti differenti diventa quindi un problema sulla media di più osservazioni univariate. Se le z sono normali, la questione può essere affrontata utilizzando un test t di Student ad un campione dove il sistema di ipotesi diventa:

$$\begin{cases} H_0 : z = 0 \\ H_1 : z > 0 \end{cases} \tag{4.6}$$

Si nota che la regione di rifiuto del test è unilaterale destra; infatti, ricordando che il vettore di differenze z è stato ottenuto sottraendo il valori di BM25 a quelli di LSPR, l'ipotesi nulla H_0 prevede che la media delle differenze sia nulla (ossia che non ci sia differenza in media fra LSPR e BM25) mentre l'ipotesi alternativa H_1 indica che il modello basato sulla trasformata discreta di Fourier ha in media valori di AP superiori rispetto alla *baseline*. Chiamando z_i i singoli valori del vettore z, resta quindi da indagare sull'ipotesi di normalità delle differenze z_i; questo, come spiegato in 4.3.1 è stato fatto sia mediante un'analisi grafica (*normal probability plot*) che con un procedimento analitico (test di normalità di Shapiro-Wilk). In figura 4.9 è riportato il *normal probability plot* relativo ai valori z_i; se i dati sono normali ci si aspetta di osservare un andamento, almeno approssimativamente, lineare. Nel caso in esame, invece, il grafico sembra suggerire un andamento non lineare e questo indica che i quantili della distribuzione dei dati non si "comportano" come quelli di una distribuzione normale ovvero

che la distribuzione dei dati non è normale. Un'ulteriore conferma in merito viene

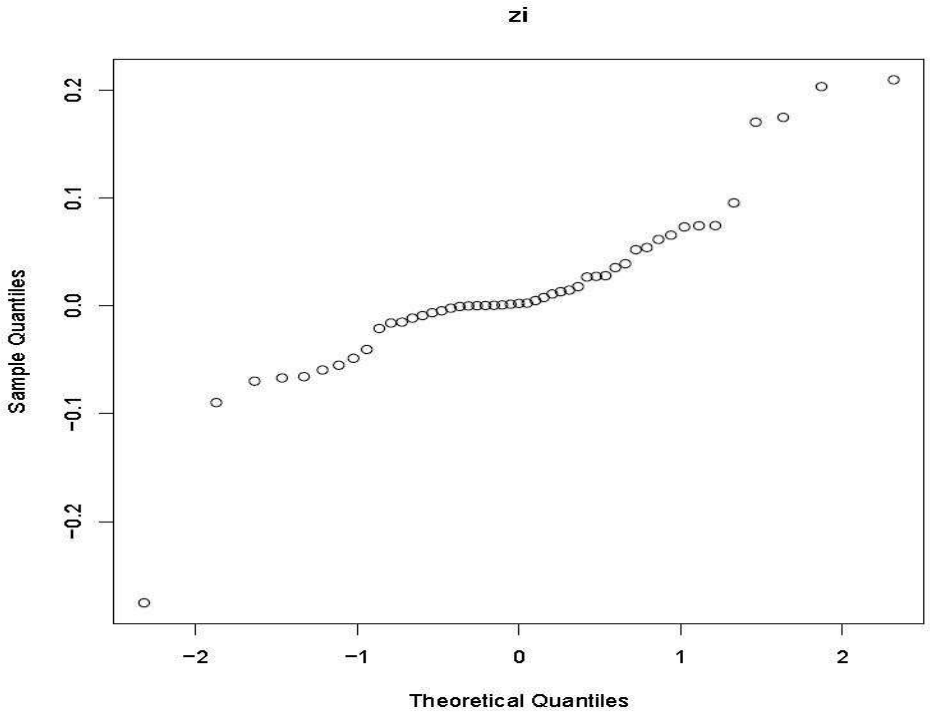


Figura 4.9: *Normal probability plot*

fornita dal test di normalità di Shapiro-Wilk il cui sistema di ipotesi è il seguente:

$$\begin{cases} H_0 : \text{la distribuzione dei dati è normale} \\ H_1 : \text{la distribuzione dei dati non è normale} \end{cases} \tag{4.7}$$

Il test si basa su una statistica che, nella sostanza, è il coefficiente di correlazione tra i punti disegnati nel *normal probability plot*; in figura 4.10 è possibile vedere i risultati dell'applicazione di tale test ai dati z_i. Il livello di significatività osservato del

```
        Shapiro-Wilk normality test

data:  zi
W = 0.8787, p-value = 0.0001187
```

Figura 4.10: Test di Shapiro-Wilk

test pari a 0.0001187, porta a dubitare fortemente sull'ipotesi di normalità dei dati in esame. Da queste considerazioni, si è deciso di procedere in ogni caso effettuando un test t di Student ad un campione ma, date le evidenze sperimentali contro l'ipotesi di

normalità delle z_i, è stato condotto anche un test non parametrico: il test di Wilcoxon. Nel seguito, quindi, verranno dapprima presentati i risultati ottenuti dall'applicazione del test t di Student ad un campione al vettore delle differenze z; successivamente si mostreranno i risultati emersi conducendo il test di Wilcoxon.

Per condurre il test t di Student ad un campione è necessario, prima di tutto, procedere con il calcolo di alcune statiche di sintesi del campione. Chiamando $\bar{z}$ la media campionaria, s^2 la varianza campionaria e n la numerosità campionaria si sono ottenuti i seguenti risultati:

$$\bar{z} = \frac{1}{n} \cdot \sum_{i=1}^{n} z_i = 0.01385 \tag{4.8}$$

$$s^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (z_i - \bar{z})^2 = 0.00602 \qquad s = \sqrt{s^2} = 0.07761 \tag{4.9}$$

Si può ora procedere al calcolo della statistica test t_{oss}:

$$t_{oss} = \frac{\sqrt{n} \cdot \bar{z}}{s} \sim t_{n-1} \qquad t_{oss} = 1.2499 \tag{4.10}$$

Per avere un'indicazione su quanto sia plausibile l'ipotesi nulla H_0 si calcola il livello di significatività osservato del test (p-value); ricordando che la regione di rifiuto è unilaterale destra, il p-value p risulta:

$$p = P(t_{n-1} \geq t_{oss}) \quad \Longrightarrow \quad p = P(t_{48} \geq 2.1677) = 0.1087 \tag{4.11}$$

Il valore di p ottenuto non permette di tratte conclusioni nette; infatti p non è così elevato da suggerire una forte evidenza a favore l'ipotesi nulla H_0. D'altro canto, non non ci sono abbastanza elementi per rigettare l'ipotesi nulla. In sostanza, quindi, siamo in una situazione di sostanziale indecisione, che a volte in statistica viene indicata come risultato *borderline*. Per cercare di avere una visione più chiara, verranno ora presentati i risultati del test di Wilcoxon, l'equivalente del test t di Student per dati appaiati, che in ambito non parametrico permette di svincolarsi da ipotesi sulla distribuzione dei dati.

Come per il test t di Student, il test di Wilcoxon non esamina i due campioni (LSPR e BM25) singolarmente ma si concentra sulla differenza tra i valori di ciascuna coppia ed il segno di ciascuna differenza; in questo test il sistema di ipotesi è il seguente:

$$\begin{cases} H_0 : \text{Mediana}(z_i) = 0 \\ H_1 : \text{Mediana}(z_i) \geq 0 \end{cases} \tag{4.12}$$

E' da notare, anche in questo caso, la regione di rifiuto unilaterale destra. Il livello di significatività osservato p del test risulta pari a 0.07773; questo valore, ancora una volta, evidenzia che ci si trova in una situazione *borderline*. Rispetto al caso precedente,

tuttavia, essendo il valore di p inferiore a 0.10, i risultati appaiono significativi al 10%, ossia il test con α=10 rifiuta H_0.

I risultati ottenuti con l'analisi di significatività, seppur non indicando una forte evidenza a favore dell'ipotesi che il modello LSPR presenti un MAP superiore rispetto al BM25, hanno messo in risalto che le differenze riscontrate nei due modelli non sembrano essere attribuite al caso. Sembra quindi che il sistema 4IR sviluppato, considerando l'indice MAP, riesca ad ottenere prestazioni confrontabili rispetto alla *baseline* BM25.

4.5 Studio di un *topic* "problematico"

Durante la valutazione di LSPR sono stati fatti diversi test variando, come spiegato, l'interrogazione fornita in ingresso per ogni *topic* e l'insieme di termini *stop word* utilizzati nell'analisi dell'esigenza informativa. Quello che emerge in tutte le prove e le misure di efficacia calcolate è che il modello basato sulla trasformata discreta di Fourier sembra avere sempre prestazioni inferiori al BM25 in alcuni *topic* della collezione sperimentale mentre, per altri, la situazione appare capovolta.

Considerato questo fatto si voleva, quindi, cercare di capire quale fosse il motivo per cui LSPR, in alcuni casi, non dava risultati soddisfacenti; per farlo si è preso in considerazione un *topic* dove le prestazioni di LSPR erano deludenti , nello specifico il numero 545, e si è fatta un'analisi spettrale. L'idea di base era la seguente: avendo a disposizione, mediante il file qrels.txt, la lista dei documenti della collezione rilevanti al *topic* e la lista dei primi 1000 documenti reperiti dal modello LSPR, si voleva fare un confronto fra lo spettro dei documenti rilevanti e quello dei documenti reperiti nelle prime posizioni dal motore in modo tale da poter capire qual'era il motivo per cui i documenti rilevanti non venivano collocati nei primi posti da LSPR. In figura 4.11 è riportato un dettaglio sul *topic* considerato, in particolare si nota la presenza, dopo la fase di analisi, di 5 termini ognuno con un diverso peso IDF. Lo spettro associato all'interrogazione prima del filtraggio è visibile in figura 4.12; per la simmetria, è stato riportato solo metà dello spettro. Come ci sia aspettava, sono presenti dei picchi intorno ai 200 punti (primo gruppo), 200+300= 500 punti (secondo gruppo), 500+300=800 punti (terzo gruppo), 800+300=1100 punti (quarto gruppo) e 1100+300=1400 punti (quinto gruppo). A questo punto è stato stato eseguito l'algoritmo LSPR e si è fatta una stampa dello spettro ottenuto, dopo il filtraggio, per tutti i documenti rilevanti e per i primi 30 documenti reperiti. Quello che si nota è che la maggior parte dei documenti nelle prime posizioni sono caratterizzati dalla presenza di tutti i termini dell'interrogazione escluso "peter" o "extend". In figura 4.13 è riportato, a titolo di esempio, lo spettro associato ad un documento in nona posizione dove si vede che sono

```
TOPIC n° 545

Interrogazione originale: to what extent did peter the great reform Russia

Interrogazione dopo l'analisi lessicale: extent peter great reform russia


Termine: extent
Termine: peter
Termine: great
Termine: reform
Termine: russia

Questa query ha 5 termini.

Il valore di N è: 4096

La frequenza del termine extent è: 401  Hz.
La frequenza del termine peter  è: 1001 Hz.
La frequenza del termine great  è: 1601 Hz.
La frequenza del termine reform è: 2201 Hz.
La frequenza del termine russia è: 2801 Hz.

Il termine extent è presente in 24584 documenti e ha peso IDF 6.10
Il termine peter  è presente in 47923 documenti e ha peso IDF 5.14
Il termine great è presente in 173049 documenti e ha peso IDF 3.29
Il termine reform è presente in 28732 documenti e ha peso IDF 5.88
Il termine russia è presente in 13322 documenti e ha peso IDF 6.99

POTENZA ASSOCIATA ALL'INTERROGAZIONE: 262106.68
```

Figura 4.11: Dettaglio del *topic* 545

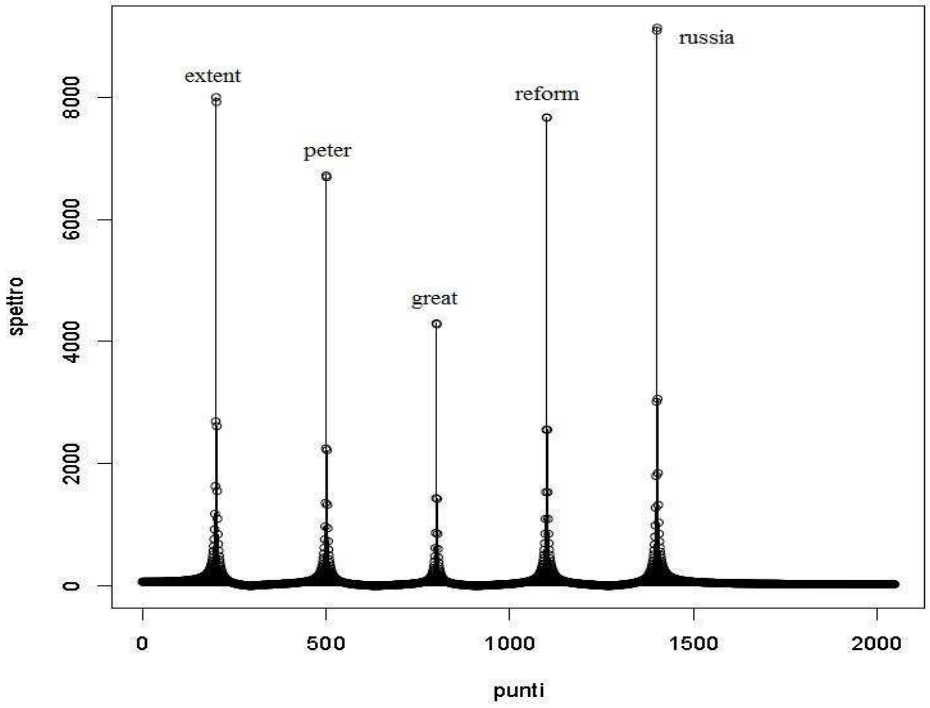


Figura 4.12: Spettro associato all'interrogazione del *topic* 545

state filtrate le componenti associate a tutti i termini dell'interrogazione ad esclusione di "peter". Per quanto riguarda i documenti rilevanti invece emerge, nella maggior

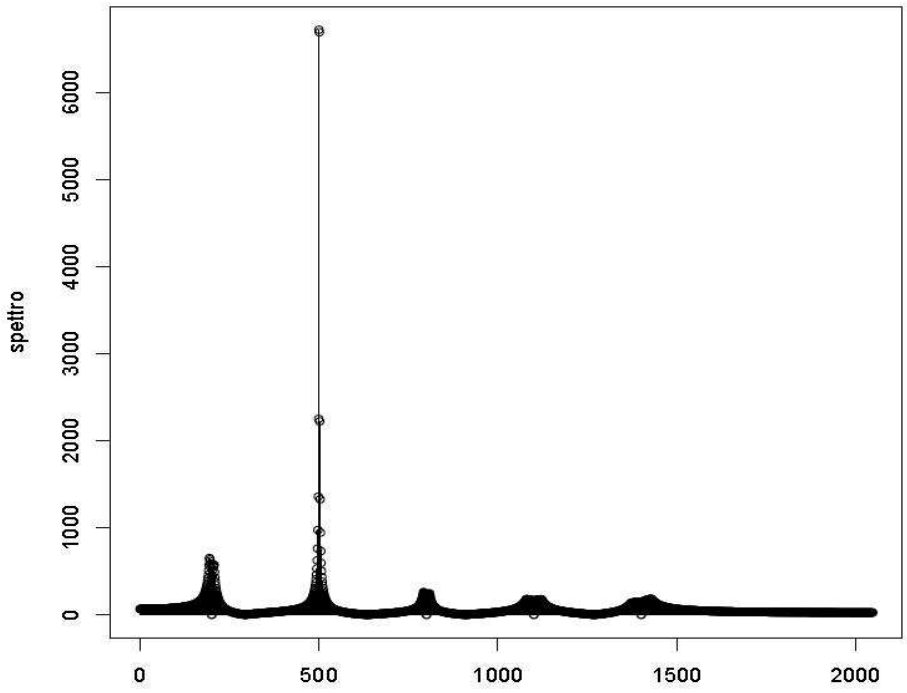


Figura 4.13: Spettro associato ad un documento in nona posizione

parte di essi, l'assenza contemporanea dei termini "extend" e "reform". In figura 4.14 viene illustrato lo spettro associato ad un documento rilevante per il *topic* 545 dove si vede che non sono state filtrate le componenti associate ai termini dell'interrogazione "extend" e "reform". Per avere una visione più completa del fenomeno, in figura 4.15 vengono riportati gli spettri associati ai primi 12 documenti rilevanti che si incontrano scorrendo i risultati di LSPR ottenuti per il *topic* 545. Questa serie di analisi grafiche dello spettro sembrano suggerire che nei documenti rilevanti il problema sia la mancanza del termine "reform"; questo causa inevitabilmente un valore maggiore dello spettro associato al documento che porta quindi LSPR a posizionarlo in fondo alla lista dei risultati. Sarebbe interessante provare, in qualche modo, a modificare il peso assegnato al termine "reform" in modo tale che la sua mancanza, all'interno dei documenti rilevanti, non comporti una penalizzazione così forte per il documento.

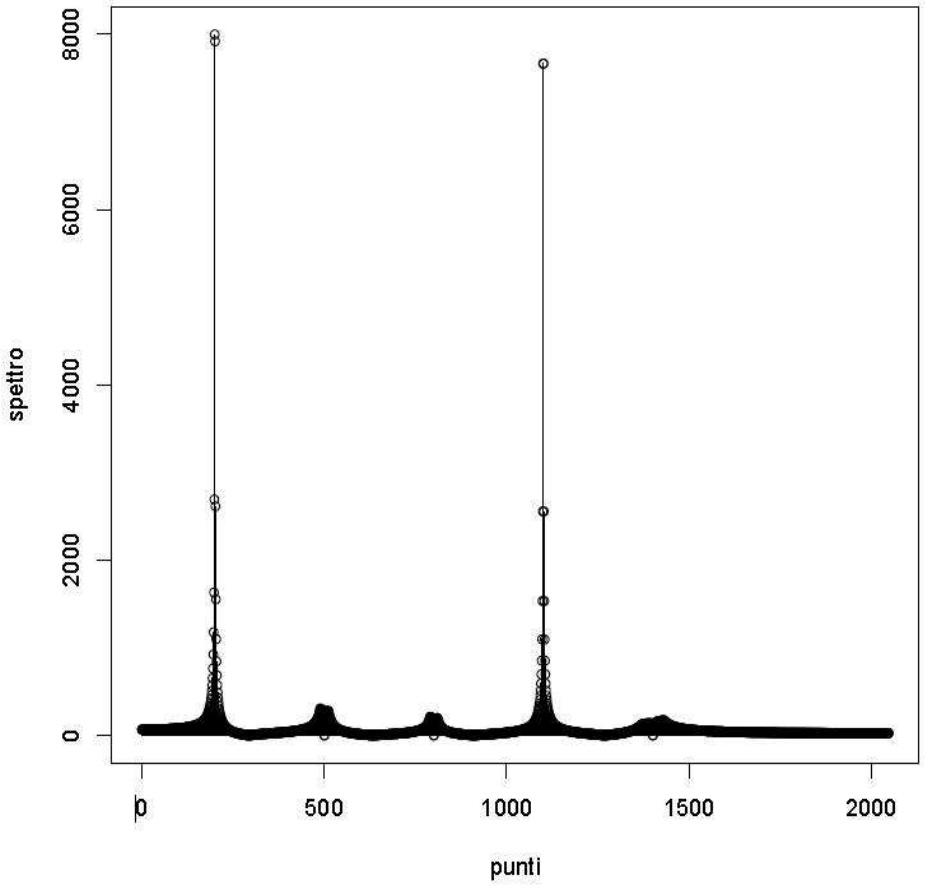
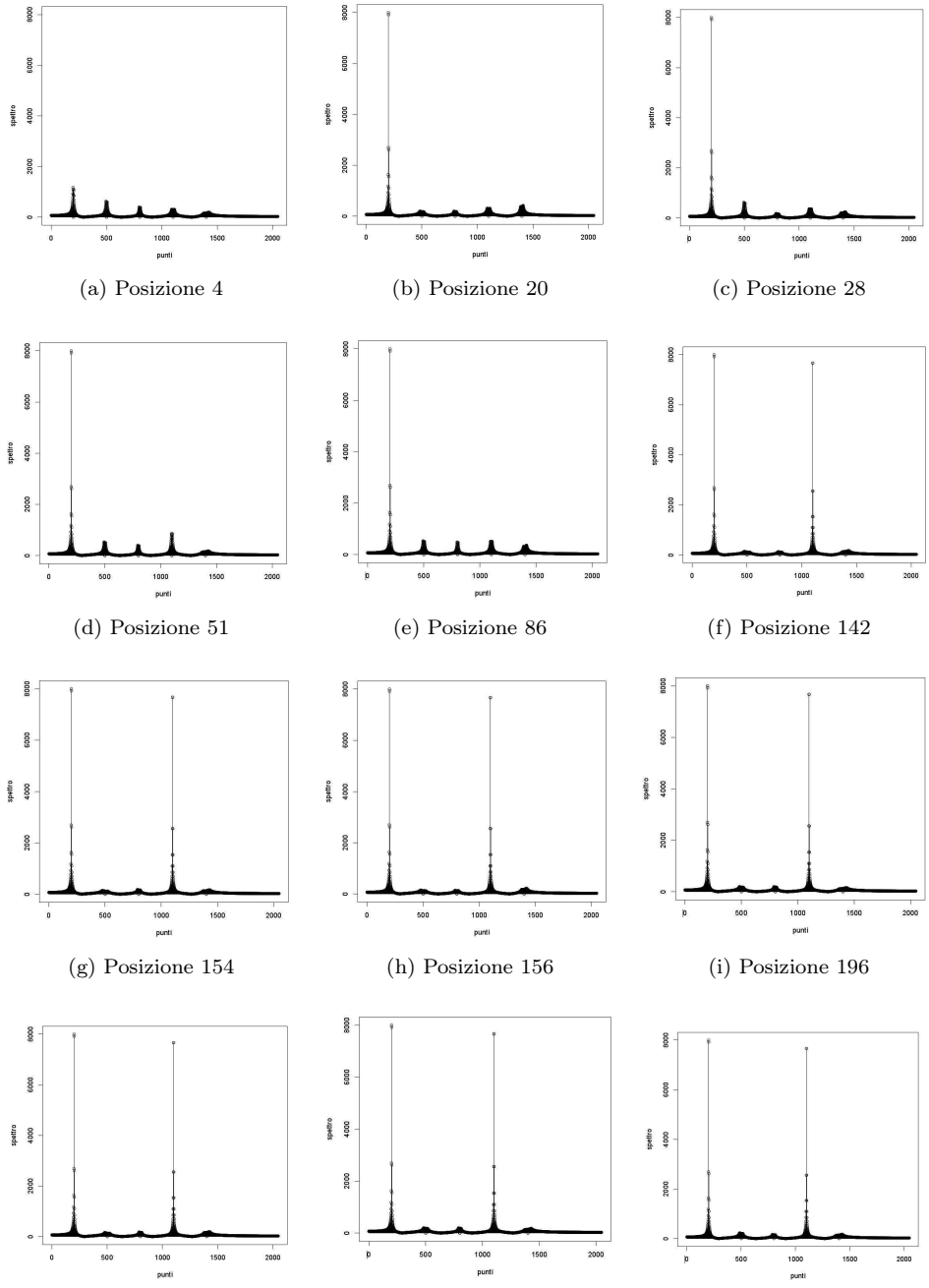


Figura 4.14: Spettro associato ad un documento rilevante



(a) Posizione 4 (b) Posizione 20 (c) Posizione 28

(d) Posizione 51 (e) Posizione 86 (f) Posizione 142

(g) Posizione 154 (h) Posizione 156 (i) Posizione 196

(j) Posizione 216 (k) Posizione 233 (l) Posizione 246

Figura 4.15: Spettro associato ai primi 12 documenti rilevanti, con relative posizioni nella lista dei risultati.

CAPITOLO

CINQUE

CONCLUSIONI

Il sistema 4IR sviluppato in questa tesi di Laurea ha avuto come punto di partenza il modello LSPR riportato in [Costa and Melucci, 2010]. Grazie all'utilizzo di strutture dati efficienti si è riusciti a realizzare un prototipo stabile, adatto a funzionare anche con collezioni sperimentali dell'ordine dei milioni di documenti. Il punto chiave di LSPR è l'utilizzo della DFT, che si è rivelata uno strumento efficace applicabile nell'ambito dell'IR. La successiva valutazione del motore sviluppato sulla collezione sperimentale "*TREC 2001 web trac ad hoc topics*" e il confronto con la *baseline* BM25 ha messo in luce alcuni punti di forza e altri di debolezza del modello basato sulla trasformata discreta di Fourier. Infatti, per alcuni *topic*, le prestazioni di LSPR appaiono superiori rispetto a BM25 che rappresenta lo stato dell'arte, mentre, in altri, la situazione si ribalta; in ogni caso, considerando come misura di efficacia il MAP, si è visto che LSPR riesce ad ottenere risultati superiori, anche se di poco, rispetto a BM25. Questa differenza è maggiormente evidenziata quando l'esigenza informativa fornita in ingresso al modello è costituita da un numero ridotto di termini; è in questo caso, infatti, che LSPR sembra avere una efficacia maggiore. Questo comportamento può trovare spiegazione nel fatto che, quando l'esigenza informativa è corta, lo spettro associato all'interrogazione presenta un numero ridotto di picchi e di conseguenza il contributo dato dal filtraggio di un picco rispetto ad un altro è elevato; quando invece l'interrogazione diventa più lunga, lo spettro ad essa associato ha un numero elevato di picchi e la differenza nel filtrarne uno piuttosto di un altro è tendenzialmente minore. LSPR può essere quindi un valido modello da applicare in ambiti come il WWW in cui le interrogazioni sono composte, nella loro stragrande maggioranza, da pochi termini.

Riguardo alla differenza di prestazioni, soprattutto in merito all'indice MAP, riscontrata nei due modelli messi a confronto, l'applicazione del test statistico t di Student per dati appaiati e del suo equivalente in ambito non parametrico, seppur non indi-

cando una forte evidenza a favore dell'ipotesi che il modello LSPR presenti un MAP superiore rispetto al BM25, hanno messo in risalto che le differenze riscontrate nei due modelli non sembrano essere attribuibili al caso. Questo risultato supporta quindi l'idea che il sistema 4IR sviluppato, considerando l'indice MAP, riesca ad ottenere prestazioni confrontabili rispetto alla *baseline* BM25.

Una direzione su cui varrebbe la pena indagare maggiormente riguarda la presenza, nella lista di risultati di LSPR per un dato *topic*, di blocchi di documenti aventi tutti uguale potenza. Questo si nota maggiormente per i documenti collocati nelle posizioni più basse della lista dei risultati. Si tratta, tuttavia, di fenomeno fisiologico derivante dal basso numero di termini delle interrogazioni.

Infine, l'idea di effettuare micro-analisi su un *topic* mediante l'analisi spettrale, ha aiutato ed evidenziare quali fossero i motivi per cui LSPR non riusciva ad ottenere buoni risultati, in termini di efficacia, per quella particolare esigenza informativa. Seguendo questa direzione, LSPR potrebbe essere ulteriormente migliorato analizzando nel dettaglio tutti i *topic* più "problematici" per capire se il malfunzionamento è in qualche modo legato da una causa comune e in tal caso ipotizzare delle soluzioni da implementare per migliorare i risultati del modello.

BIBLIOGRAFIA

[Briggs et al., 1995] Briggs, W. et al. (1995). *The DFT: An Owner's Manual for the Discrete Fourier Transform*. Society for Industrial and Applied Mathematics, Philadelphia.

[Costa, 2009] Costa, A. (2009). Lspr: un modello di reperimento dell'informazione. Tesi di Laurea Specialistica, Università degli Studi di Padova.

[Costa and Melucci, 2010] Costa, A. and Melucci, M. (2010). An information retrieval model based on discrete fourier transform. In Cunningham, H., Hanbury, A., and Rüger, S., editors, *Advances in Multidisciplinary Retrieval*, volume 6107 of *Lecture Notes in Computer Science*, pages 84–99. Springer Berlin / Heidelberg.

[Costa and Roda, 2011] Costa, A. and Roda, F. (2011). Recommender systems by means of information retrieval. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 57:1–57:5, New York, NY, USA. ACM.

[Dym and McKean, 1972] Dym, H. and McKean, H. P. (1972). *Fourier Series and Integrals*. Academic Press, New York. 129 pp.

[Grossman, 2005] Grossman, D. (2005). Notes on trec eval.

[Hatcher et al., 2010] Hatcher, E., Gospodnetic, O., and McCandless, M. (2010). *Lucene in Action*. Manning, 2nd revised edition. edition.

[Oppenheim et al., 1996] Oppenheim, A., Willsky, A., et al. (1996). *Signal & system*. Prentice-Hall, 2nd edition.

[Robertson and Zaragoza, 2009] Robertson, S. E. and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.