Reperimento dell'Informazione – Homework 1

Studente: Davide Masiero
Matricola: 1179783
Repository: https://github.com/XWerer/IR_Homework

Introduzione

Il seguente homework consiste nell'usare un sistema di IR, che nel seguente caso sarà Terrier alla versione 4.4, per andare ad indicizzare una collezione sperimentale TREC7, di circa 528000 documenti, contenente 50 differenti topic. Si andrà quindi a creare 4 diversi indici (uno per ogni run), su cui poi si andranno a fare la parte di reperimento e successivamente quella di valutazione (calcolando nello specifico MAP, RPrec e Precision at 10), utilizzando un pool con due gradi di rilevanza: R (rilevanti) e NR (non rilevanti). Le run quindi avranno le seguenti caratteristiche:

- Stoplist, Porter stemmer, BM25 (SL_PS_BM25).
- Stoplist, Porter stemmer, TF*IDF (SL_PS_TF_IDF).
- No stoplist, Porter Stemmer, BM25 (PS_BM25).
- No stoplist, No stemmer, TF*IDF (TF_IDF).

Successivamente sì andrà a svolgere un test statistico, ANOVA 1-way, per determinare i sistemi appartenenti al "top group" sulla base delle diverse misure.

Indicizzazione, Reperimento e Valutazione

Come prima cosa è stata fatta l'indicizzazione dei documenti, una per ogni run, andando a settare propriamente il file *properties* usato da Terrier. Qui mi sono scontrato con il dover fare le prime scelte progettuali per ogni sistema, che poi sono andate a influire sui risultati ottenuti nella fase di valutazione. Infatti, Terrier mette a disposizione molti parametri con cui poter variare il proprio sistema IR. I parametri che mi sono concentrato a modificare sono due: *ignore.low.idf.terms* e *TrecQueryTags.process.* Che indicano rispettivamente, se nella fase di reperimento, vanno ignorati i termini con valori bassi di IDF (valori molto bassi indicano che il termine è molto frequente nella collezione quindi magari meno rilevante), e che parte processare dei vari topic (in questo caso se considerare solo il *title* o anche la *desc*). Ho eseguito varie prove per ogni run, e ho deciso di tenere in considerazione solo quelle che mi hanno portato ad avere resultati migliori nella fase di valutazione, poiché il tag *TrecQueryTags.process* è stato settato in modo tale da considerare sempre sia il *title* e la *desc*, infatti la parte descrittiva del topic (*desc*), contiene molte più parole chiave rispetto al solo utilizzo del titolo (*title*), e questo a portato ad un incremento delle prestazioni dei vari sistemi. Mentre il tag *ignore.low.idf.terms* e stato settato appositamente ad ogni run per ottenere il miglior risultato. Nel repository (link sopra riportato), troverete anche il file *properties* utilizzato per ogni run.

Dopo aver creato l'indice, si è passati alla fase di retrieval, che Terrier svolge in automatico tramite un semplice comando da terminale (come la fase di indicizzazione), fornendogli in input un file contenente i topic. Successivamente si è andati a svolgere la valutazione, anche questa implementata con un semplice comando da terminale, anche se ho preferito ottenere ulteriori informazioni andando ad usare il tool *trec_eval*. Di seguito è presente una tabella riassuntiva con i valori medi ottenuti, mentre nel repository sono presenti 3 grafici che comparano le 3 misure effettuate, prodotti usando del codice scritto in python, anche questo presente nel repository.

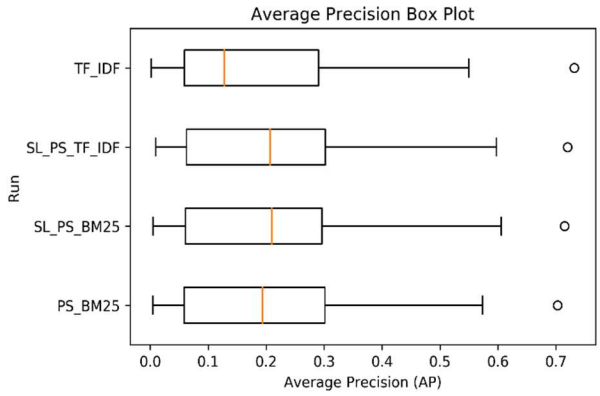
	MAP	RPrec	P_10
SL_PS_BM25	0.2125	0.2705	0.4820
SL_PS_TF_IDF	0.2123	0.2725	0.4780
PS_BM25	0.2108	0.2740	0.4740
TF_IDF	0.1876	0.2485	0.4260

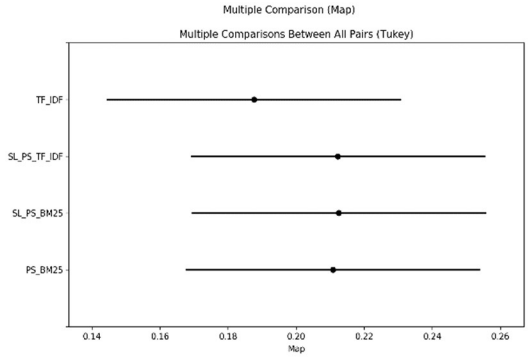
Dai seguenti dati possiamo dedurre che i primi 3 sistemi sono i migliori, infatti hanno ottenuto risultati molto simili tra loro, mentre possiamo vedere come l'ultimo sistema che non usa né il Porter Stemmer né la rimozione delle Stop Word abbia avuto difficolta, inoltre ha avuto risultati migliori settando il flag *ignore.low.idf.terms* a *false*, probabilmente a causa del fatto che non usa il Porter Stemmer.

Test Statistico ANOVA 1-WAY

L'ultima parte di questo homework prevede lo sviluppo del test statistico ANOVA 1-WAY. Il test è stato svolto creando un piccolo programma in python, disponibile anch'esso nel repository. Di seguito si trovano una tabella che raccoglie i valori P ed F per ogni valore calcolato in precedenza (Average Precision indicato con il nome di MAP, RPrec e P_10), il BoxPlot e il grafico rappresentante il test di Tukey HSD solo sull'AP (disponibili anche gli altri BoxPlot e test di Tukey nel repository, mostro solo questi poiché sono i più significativi).

	P	F
MAP	0.851	0.264
RPrec	0.836	0.284
P_10	0.759	0.390





Come si può vedere dai valori abbastanza alti di P, la probabilità che l'ipotesi nulla H0 (medie tutte uguali), sia verificata è alta, un ulteriore conferma di ciò ce la da il test di Tukey riportato in figura (nel codice si vede anche la tabella), che tutti i sistemi IR sviluppati appartengono al top group, con i primi 3 sistemi (come visto anche in precedenza), con prestazioni simili (nel grafico sono gli ultimi 3), mentre il sistema con il solo modello TF_IDF senza Stop List e Porter Stemmer fa fatica.