Homework 1 - Reperimento dell'Informazione - A.A. 18/19

Eugen Saraci - 1171697
Università degli Studi di Padova
eugen.saraci@studenti.unipd.it

I. INTRODUZIONE

L'homework consiste nel confronto fra diverse configurazioni di un sistema di reperimento differenziati principalmente dal modello di retrieval e dalle diverse pipeline di preprocessing utilizzati. Il corpus dei documenti è rappresentato dalle collezioni sperimentali Disks4&5 di TREC7, consistenti in circa 528.000 documenti di cui abbiamo a disposizione 50 topic con giudizi di rilevanza binari. L'obiettivo è quindi utilizzare un qualsiasi sistema di reperimento citato a lezione ed analizzare le performance delle varie configurazioni richieste, determinando, ad esempio, quale sia la migliore. Le varie configurazioni (che chiamerò per semplicità *sistemi*[1]) sono le seguenti:

Model	Pipeline steps	Short name
BM25	PorterStemmer, Stopwords	`bm25_full`
TF IDF	PorterStemmer, Stopwords	`tf_idf_full`
BM25	PorterStemmer	`bm25_nostop`
TF IDF	–	`tf_idf_none`

TABLE I. Le quattro configurazioni richieste dall'homework. La colonna "short name" contiene i nomi con cui si farà riferimento ai singoli sistemi.

Il sistema di reperimento che ho deciso di utilizzare per l'homework è Terrier (ver. 4.4) [1], mentre la libreria utilizzata per l'evaluation è `trec_eval` [2]; per i test statistici sono stati utilizzati i moduli Python `scipy.stats` [3] e `statsmodels` [4].

A. Organizzazione della relazione

Nella Sezione II presento brevemente il software siluppato; nella Sezione III fornisco una descrizione di quelli che sono stati i passaggi fondamentali che hanno portato al completamento dell'homework.

II. SOFTWARE SVILUPPATO

Terrier offre una ampia possibilità di scripting, per tale motivo ho deciso di sviluppare degli script in Bash e Python al fine di automatizzare il processo che va dal preprocessing iniziale dei documenti fino alla stampa dei plot dei test statistici, coprendo tutti i passaggi richiesti dall'homework. Il software sviluppato, seppur commentato e documentato con cura, non è orientato al riuso, è bensì un modo veloce che permette di replicare con semplicità i risultati ottenuti. L'utente interessato alla replicazione dei risultati deve solamente scaricare il corpus dei documenti (ovvero la cartella TIPSTER), spostarla all'interno delle cartelle del progetto come specificato nelle istruzioni, e solo a quel punto dovrà eseguire lo script denominato `main.sh`.

La documentazione completa, le istruzioni per l'utilizzo, ed il software stesso sono reperibili alla repository presente al link `https://github.com/esaraci/IR-HW1`. Nella repository sono presenti anche tutte le tabelle e le immagini generate dagli script, molte delle quali non sono riportate in questo elaborato.

III. FASI OPERATIVE

L'homework è suddivisibile, secondo personale interpretazione, in cinque fasi distinte: *Preprocessing*, *Indexing*, *Retrieval*, *Evaluation*, *Hypothesis Testing*.

A. Preprocessing

Nella fase di preprocessing vengono creati i file e le cartelle necessari alla corretta esecuzione delle fasi seguenti. Non essendo una fase strettamente legata allo scopo della relazione ne cito solo i passaggi che ritengo rilevanti. Uno di questi è la rinominazione dei file con estensione `.{1,2,3}Z`. I file appena citati non sono manipolabili dal comando POSIX `uncompress` solamente a causa della loro estensione (non per il formato), motivo per cui li ho rinominati in `_{1,2,3}.Z`, in maniera tale da essere accettati senza errori o warning da `uncompress`.

In alcuni file sono presenti dei commenti, essi possono essere rimossi attraverso alcuni comandi, tuttavia ho verificato empriricamente che la loro rimozione non porta ad alcun miglioramento nelle performance; ho deciso di tenerli.

Rientra in questa fase anche la parte di ricerca relativa alla miglior configurazione per il file `terrier.properties`. La filosofia seguita per la scrittura del file è stata quella di introdurre in esso il minimo necessario per il corretto e buon funzionamento di Terrier. Ad esempio non sono stati definiti esplicitamente i parametri il cui valore predefinito fosse quello voluto[2]; il modello di reperimento, il nome degli indici, i file di output non sono stati inseriti nel file in quanto vengono specificati automaticamente dal software introdotto alla Sezione II. Alcuni tag sono stati inseriti basandosi su [6].

B. Indexing

In questa fase vengono creati gli indici richiesti. Si noti che sebbene i sistemi siano 4, gli indici sono solamente 3, ciò si può intuire dalla Tabella I in quanto sono solo i passi

[1]Con sistema intendo la coppia composta dal modello di reperimento e dalla pipeline. Da non confondere con la parola sistema intesa ad indicare Terrier nel suo complesso.

[2]Eccezion fatta per quelli il cui cambiamento risultasse comodo per esperimenti veloci.

della pipeline ad influire sull'indice e ci sono solo 3 diverse pipeline. Osservando le dimensioni dei *lexicon* ho notato che l'applicazione del PorterStemmer arriva a rimuovere circa 100.000 termini dagli 840.000 iniziali, mentre la rimozione delle stopword porta un ulteriore decremento di 200 termini.

C. Retrieval

La fase di retrieval è la fase in cui Terrier, ricevendo in input le query (o *topic*) a nostra disposizione, restituisce una lista di documenti ritenuti rilevanti per le query. Questa fase è svolta su tutti e 4 i sistemi, portandoci ad avere risultati (o *run*) possibilmente diversi per ogni topic e per ogni sistema. La valutazione della bontà delle run si svolge nella fase di *evaluation*.

D. Evaluation

L'evaluation consiste nel confrontare le run ottenute dalla fase di *retrieval* con i giudizi di rilevanza binari a nostra disposizione. Riporto in forma tabellare (Tabella II) e grafica (Figura 1) alcuni dei risultati essenziali.

System	MAP	RPrec	P@10
`bm25_full`	0.1828	0.2391	0.418
`tf_idf_full`	0.1821	0.2391	0.42
`bm25_nostop`	0.1857	0.2409	0.43
`tf_idf_none`	0.1693	0.229	0.406

TABLE II. Medie dei valori ottenuti sui 50 topic. Sono state riportate solo le metriche richieste dall'homework.

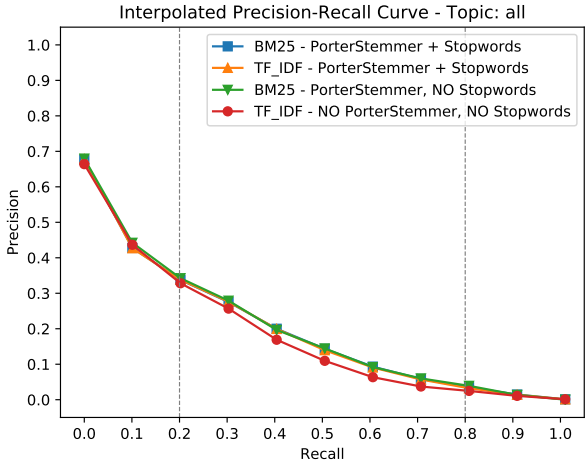

Fig. 1. Valori (interpolati) medi sui 50 topic di *precision* a diversi livelli di *recall*. L'andamento della curva è prevedibile in quanto al crescere del numero di documenti reperiti aumenta anche il numero di documenti non rilevanti reperiti, facendo così diminuire la *precision* [5]. Le linee grigie veritcali facilitano il confronto fra i sistemi delimitando le tre aree in cui risulta essere più interessante compararli [5]. Nessun sistema sembra essere notevolmente migliore rispetto agli altri.

E. Hypothesis Testing

I risultati ottenuti in fase di evaluation non sono sufficienti a determinare quale modello sia significativamente il migliore; per provare a fare ciò è necessario ricorrere a dei test statistici. Usiamo one-way ANOVA (Tabella III) per determinare se almeno uno dei sistemi risulti essere significativamente diverso dagli altri, successivamente, per individuare tale sistema (o sistemi), facciamo un confronto *pairwise* fra di essi attraverso il test di Tukey (Figura 2).

Measure	F-stat	p-value
MAP	0.0997	0.9601
RPrec	0.0567	0.9822
P@10	0.0446	0.9875

TABLE III. I valori della colonna *p-value* indicano la probabilità di osservare misure simili in presenza dell'ipotesi nulla. Essendo alti non abbiamo prove empiriche per rigettare tale ipotesi, la quale sostiene che le misure ottenute nei 50 topic dai 4 sistemi provengano dalla stessa distribuzione di probabilità, in breve: nessuno dei sistemi è signficativamente diverso dagli altri (a livello di signficatività 0.05).

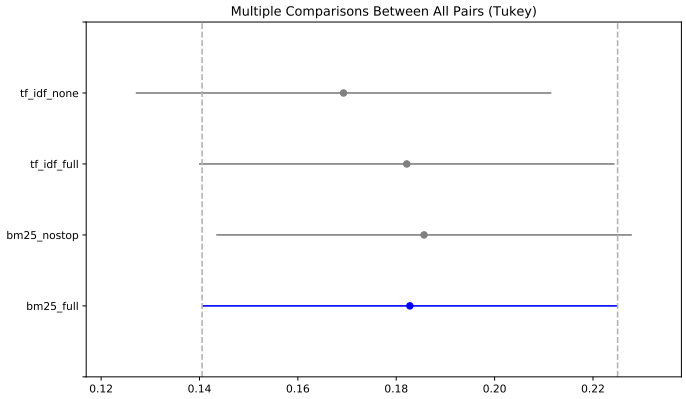

Fig. 2. In blu è colorato l'intervallo di confidenza del sistema che pensavo raggiungesse le migliori performance (`bm25_full`); in grigio sono colorati i sistemi NON significativamente diversi dal sistema blu; in rosso (nessuno in figura) sono colorati i sistemi significativamente diversi, ovvero quelli con intervalli di confidenza disgiunti dal sistema blu. Il grafico coferma nuovamente ciò che già avevamo visto con ANOVA, ovvero che non abbiamo prove empiriche per rigettare l'ipotesi nulla.

IV. NOTE AGGIUNTIVE

In [7] sono riportati i risultati ottimali ottenuti da Terrier sulle collezioni a nostra disposizione. Essi sono stati raggiunti cambiando il valore del parametro `b` del modello *BM25*, operazione che non mi è possibile fare da riga di comando. I valori riportati in questo elaborato possono essere migliorati aggiungendo il tag `DESC` al parametro `TrecQueryTags.process` all'interno del file `terrier.properties`, ma ciò non è usuale in quanto il contenuto del tag `DESC` è solo una forma d'aiuto verso i *tutor*, non fa quindi parte del contenuto della query.

REFERENCES

[1] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006). 10th August, 2006. Seattle, Washington, USA

[2] https://trec.nist.gov/trec_eval/index.html

[3] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001, http://www.scipy.org/

[4] Seabold, Skipper, and Josef Perktold. “Statsmodels: Econometric and statistical modeling with python.” Proceedings of the 9th Python in Science Conference. 2010.
[5] https://trec.nist.gov/pubs/trec16/appendices/measures.pdf
[6] http://ir.dcs.gla.ac.uk/wiki/Terrier/Disks4&5
[7] http://terrier.org/docs/v4.0/trec_examples.html