Reperimento dell'informazione - Homework 1

Ylenia Grava (1179778)

URL del progetto: `https://github.com/yleniag/Information-Retrival-Homework-1`
Contenuto della repository: le quattro run, gli output delle valutazioni, il Jupyter notebook per l'estrazione e l'analisi delle misure e i plot.

Indicizzazione, produzione delle run e valutazione

Come sistema di reperimento è stato utilizzato Terrier (v. 4.4) in modalità batch su sistema operativo Windows 10. Con Terrier sono state quindi eseguite l'indicizzazione della collezione TREC messa a disposizione (per un totale di 528155 documenti) e la produzione delle quattro diverse run richieste.
Tramite il comando `trec_setup.bat` è stato indicato a Terrier il path della cartella contentente la collezione TREC Tipster da indicizzare. Prima di creare l'indice (`trec_terrier.bat -i`), si è modificato il parametro `termpipelines` nel file `trec.properties`, specificando se si desiderasse indicizzare documenti e topic con Porter stemmer e/o rimozione delle stopword (utilizzando la stop list e il Porter stemmer di default di Terrier). Si è inoltre indicato, aggiungendo il parametro `trec.topics`, dove trovare il file contenente i topics in formato TREC (`topics.351-400_trec7.txt`) e si sono modificati i parametri `TrecQueryTags.process` e `TrecQueryTags.skip` specificando quali tag dei topic utilizzare, ovvero titolo e descrizione, e quali tralasciare, ovvero il "narrative", per evitare di appesantire computazionalmente anche se a leggero discapito della precisione che si andrà a ottenere. Si sono quindi ottenuti tre indici differenti da utilizzare nella produzione delle diverse run.
Dopo aver eseguito l'indicizzazione si è potuto passare alla fase di retrieval. Si sono quindi prodotte le quattro run richieste (`file.res`) tramite il comando `trec_terrier.bat -r -Dtrec.model= 'MODELLO'`, specificando se utilizzare il `BM25` o il `TF_IDF`. Si è deciso di ignorare le parole con valori di IDF basso, quindi le parole troppo frequenti (oltre alle stopword nel caso in cui vengano rimosse), modificando il parametro `ignore.low.idf.terms=true`. L'indice utilizzato per creare la run viene automaticamente recuperato dall'apposita cartella `index`, è stato necessario quindi prestare attenzione a utilizzare l'indice corretto a ogni run prodotta. Le run ottenute sono le seguenti:

- Run0: Porter stemmer, rimozione stopword, BM25;
- Run1: Porter stemmer, rimozione stopword, TF_IDF;
- Run2: Porter stemmer, BM25;
- Run3: nè Porter stemmer nè rimozione stopword, TF_IDF.

La valutazione delle run è stata condotta utilizzando la libreria trec_eval, messa a disposizione da Terrier. Tramite il comando `trec_eval.bat -q -m all_trec path/to/qrels.txt path/to/run.res` (con `qrels.txt` il file contenente i giudizi di rilevanza binaria) sono state valutate le run, ottenendo una serie di misure per ognuno dei 50 topic. Trec_eval calcola anche le medie dei 50 valori per ogni misura (a cui fa riferimento con la dicitura topic "all"). Si sono quindi raccolti tutti questi risultati in quattro file di testo, uno per ogni run e si è poi scritto un programma in Python 3.7 con l'utilizzo di Jupyter Notebook, col fine di raccogliere tramite un parsing e plottare per tutte le diverse run i quattro valori di *Mean Average Precision* (MAP), il valore di *Average Precision* (AP, ma chiamata map nell'output di trec_eval), *Precision at 10* (P10) e *Precision at Recall Base* (rPrec) per ogni topic. Come ci si poteva aspettare, dai risultati ottenuti si nota che il valore medio di P10 è più alto rispetto a quello di rPrec e AP, indicando che i sistemi ottengono una maggiore precisione considerando i primi 10 documenti più rilevanti. Questo si può notare anche plottando i singoli valori delle misure per ogni topic: l'istogramma della P10 ottiene valori genericamente più alti, ci sono infatti molti più topic con precisione pari a 1, cosa che non avviene mai con le altre due misure (i plot sono disponibili nella repository Github).
Dal plot dei quattro valori di MAP (Figura 1) si può notare che l'ultima run ottiene il risultato peggiore (18.7% circa di precisione), mentre le prime due run ottengono risultati molto simili tra loro (21.2% circa).
Sono state infine analizzate le distribuzioni di dati ottenute. Dopo aver visualizzato i dati tramite boxplot, sono stati condotti i test statistici ANOVA 1-way e Tukey HSD test. Nella stesura del codice Python sono state utilizzate le seguenti librerie: `numpy` per il supporto a vettori, matrici e funzioni matematiche, `matplotlib` per i plot, `scipy.stats` e `statsmodels` per il supporto ai test statistici.

ANOVA 1-WAY, Tukey HSD test

Si è condotto il test statistico ANOVA 1-way in Python, in un primo momento considerando solo la misura AP, ponendo il valore di soglia `alpha` pari a 0.05. La null hypothesis (H_0) nel nostro caso consiste nell'affermare che le medie delle distribuzioni dei dati presi in considerazione non sono significativamente differenti statisticamente. Il `p-value` risultante (Tabella 1) è molto maggiore di `alpha`: falliamo quindi nel rifiutare la null hypothesis. Si è condotto quindi il Tukey HSD test, che mette maggiormente in evidenza eventuali differenze. I risultati sono riportati in Tabella 2, dove vengono confrontate tutte le possibili coppie di run, e in Figura 2, dove è mostrato il confronto tra tutte le run, con la media e l'intervallo di confidenza di ciascuna.
Notando, come già detto in precedenza, che i valori della P10 relativi ai diversi topic sono generalmente più alti, si è deciso di condurre gli stessi test anche con le misure P10 e rPrec per valutare se i risultati ottenuti con l'AP venissero confermati o meno. Si è quindi constatato che per tutte e tre le misure si fallisce nel rigettare la null hypothesis, che tutte le run rientrano nel top group (come si può vedere nelle Figure 3 e 4 esiste sempre un'intersezione non nulla tra i quattro intervalli di confidenza), ma che la Run3 risulti sempre avere valori di precisione più bassi rispetto alle altre.

Plot e tabelle

Di seguito sono riportati alcuni dei plot e tabelle, tutti gli altri grafici sono disponibili nella repository Github.

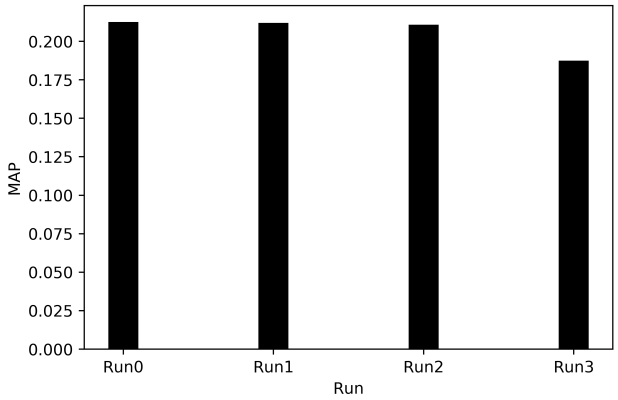

Figura 1. Valori di MAP per ogni run

	AP	P10	rPrec
F-value	0.2698	0.3578	0.3508
p-value	0.8471	0.7836	0.7886

Tabella 1. Anova 1-way con le tre diverse misure

gruppo 1	gruppo2	meandiff	lower	upper	reject
Run0	Run1	-0.0005	-0.0865	0.0855	False
Run0	Run2	-0.0018	-0.0877	0.0842	False
Run0	Run3	-0.0251	-0.1111	0.0609	False
Run1	Run2	-0.0012	-0.0872	0.0848	False
Run1	Run3	-0.0246	-0.1106	0.0614	False
Run2	Run3	-0.0233	-0.1093	0.0626	False

Tabella 2. Tukey HSD test (AP): confronto a coppie

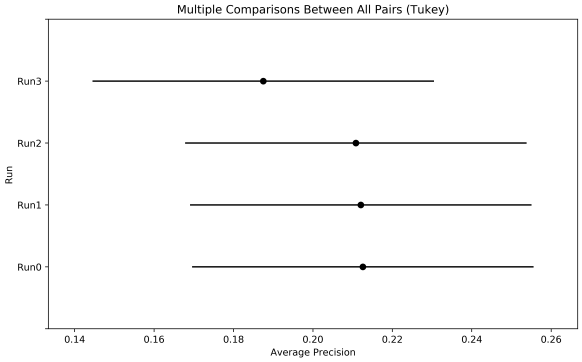

Figura 2. Tukey HSD Test (AP): confronto multiplo

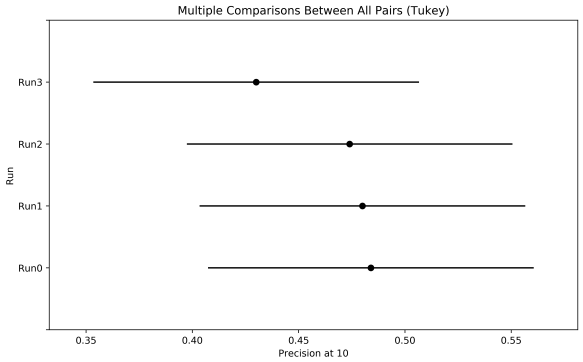

Figura 3. Tukey HSD Test (P10) : confronto multiplo

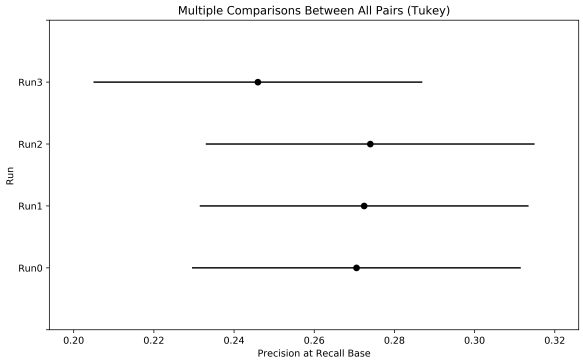

Figura 4. Tukey HSD Test (rPrec): confronto multiplo

Conclusioni e considerazioni finali

A partire dai grafici relativi alle misure per i singoli topic, si è notato come in generale tutte le run ottengano una precisione maggiore per i primi 10 documenti rilevanti (ottenendo precisioni pari a 1 per diversi topic), mentre con le altre misure si raggiunge al massimo un valore di precisione di 0.7 per il topic numero 365. Inoltre, si è notato che i risultati sulla MAP delle prime due run, condotte indicizzando con Porter stemmer e rimozione delle stopword, sono migliori. La media da sola non è così significativa per evidenziare differenze su distribuzioni di dati, pertanto sono stati condotti i test statistici. L'ANOVA e il Tukey HSD test ci permettono di concludere a loro volta che le quattro run ottengono risultati simili poichè appartengono tutte al top group, ma le prime due sono migliori e performano in maniera molto simile. La scelta tra BM25 e TF_IDF non impatta quindi sui risultati. La Run3 si conferma essere quella che ottiene i risultati peggiori, risentendo del fatto che l'indicizzazione è stata condotta senza stemmer e senza rimozione delle stopword: questo incide sia sull'efficacia, per il livello di precisione raggiunta, che sull'efficienza, poichè il tempo per produrre ed analizzare l'ultimo indice (essendo anche di maggiori dimensioni) è naturalmente risultato essere leggermente maggiore rispetto ai due precedenti.