

Homework 1: Relazione

Francesco Brigo, 1179345

11 gennaio 2019

1 Configurazioni utilizzate

La collezione utilizzata è TIPSTER: TREC 7 vol.4 & 5.

Sistema di IR: Terrier v.4.4

Sistema di valutazione: Trec eval, contenuta all'interno di Terrier

Linguaggio utilizzato: Python

Librerie utilizzate: Matplotlib, Tabulate, Scipy, Statsmodels e Numpy

Identificazione delle run:

- RUN#0: modello BM25 con PorterStemmer e Stopwords;
- RUN#1: modello TFIDF con indice con PorterStemmer e Stopwords;
- RUN#2: modello BM25 con PorterStemmer;
- RUN#3: modello TFIDF.

Link a repository: <https://github.com/FrancescoBrigo/IR-HW1>

2 Impostazione della collezione, creazione degli indici, delle run e valutazione

Il sistema di IR Terrier permette di impostare la collezione con il comando:

```
trec_setup.bat Path\To\TIPSTER
```

Una volta creata la collezione si procede alla creazione degli indici con il comando:

```
trec_terrier.bat -i -Dtermpipelines=Stopwords,PorterStemmer
```

dove *-Dtermpipelines* indica, nel nostro caso, la presenza o meno di stoplist e stemmer per le differenti run.

Una volta creati i diversi indici si modifica il file *terrier.properties* come segue:

- 1 si inseriscono i topic aggiungendo la riga:

```
trec.topics=Path\To\Topics;
```

- 2 si inserisce il comando *ignore.low.idf.terms=true*, in modo da scartare i termini con bassa IDF;

- 3 si imposta il campo: *TrecQueryTags.process=TITLE, DESC* in modo da processare sia titolo che la descrizione per rendere le query più attinenti al documento in formato TREC, confidando, di conseguenza, in risultati migliori rispetto al processamento di un solo campo;

- 4 si imposta il campo *TrecQueryTags.skip=NARR* in modo da non processare la narrazione.

Ora si eseguono le run con il comando:

```
trec_terrier.bat -r -Dtrec.model=MODELLO
```

dove MODELLO può essere BM25 o TF_IDF, a seconda del modello scelto.

Come già detto, la valutazione viene effettuata con *trec_eval* e viene eseguita con questo comando, in cui RUN.res sono i risultati del precedente comando.

```
trec_eval -q -m all_trec Path\To\Qrels\qrels.trec7.txt  
Path\To\Results\results\RUN.res »  
./../var/valutazioni/valutazioneRUN.txt
```

A questo punto si avranno i file delle valutazioni che conterranno tutte le misure necessarie. Nel codice disponibile nel [repository allegato](#) viene effettuato un parsing del file di valutazione in modo da ottenere, nei rispettivi array, i dati necessari per poter effettuare i test statistici ANOVA e TukeyHSD (i cui risultati sono mostrati successivamente in fig.5 e fig.6).

Dai test effettuati è risultato che:

- in riferimento all'ANOVA sono risultati pvalue ~0.80, facendo dunque accettare la *null-hypothesis* e riportando quindi una forte similarità fra le varie run;
- in riferimento al TukeyHSD test risultano evidenti le differenze minime fra i gruppi, anche se fanno tutti parte del top group, e viene evidenziato la top-run, che nel caso delle misure AP e P@10 risulta essere la RUN#0, mentre per Rprec risulta essere la RUN#2;
- la run migliore, con riferimento alle metriche MAP, Precision@10, è quella effettuata con modello BM25 e con Porter Stemmer e Stoplist. Tuttavia la Rprec migliore si ha per la RUN#2 ossia quella effettuata con modello BM25 e il solo Porter Stemmer;
- le run effettuate con i modelli BM25 e TF_IDF con Porter Stemmer e Stoplist hanno risultati molto simili;
- la run con modello TF_IDF senza stemmer e stoplist è risultata, abbastanza intuitivamente, la peggiore;

Globalmente, i risultati vedono una Average Precision abbastanza bassa, con però un notevole incremento con l'utilizzo dello stemmer e della stoplist (dal 18% e al 21%), una Precision@10 media di circa il 46% (ed un incremento dal 43% al 48%) e una Rprec media di circa il 26% (ed un incremento dal 24% al 27%).

Si può inoltre aggiungere che, nel caso di utilizzo del modello TF_IDF, la valutazione è molto influenzata dalla presenza (o meno) dello stemmer, sempre PorterStemmer nel caso in esame, e della stoplist, tanto che la run 3 è risultata la peggiore in ogni parametro di confronto con le altre.

Ulteriori grafici sono disponibili nel repository allegato.

RUN ID	MAP	Precision@10	Rprec
RUN#0	0.2126	0.484	0.2705
RUN#1	0.212	0.48	0.2725
RUN#2	0.2108	0.474	0.274
RUN#3	0.1875	0.43	0.246

Figura 1: Le misure ottenute per ogni run

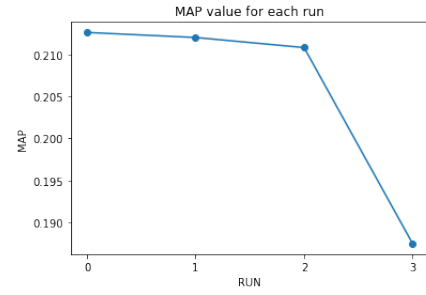


Figura 2: MAP per ogni run

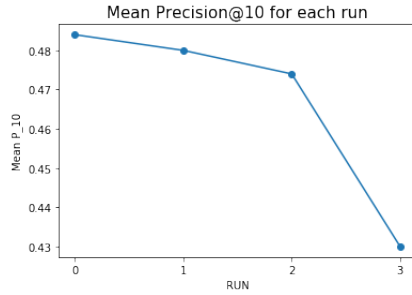


Figura 3: P@10 per ogni run

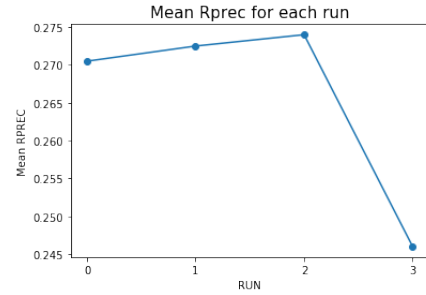


Figura 4: Rprec per ogni run

```

Multiple Comparison of Means - Tukey HSD,FWER=0.05
=====
group1 group2 meandiff lower upper reject
-----
RUN#0 RUN#1 -0.0005 -0.0865 0.0855 False
RUN#0 RUN#2 -0.0018 -0.0877 0.0842 False
RUN#0 RUN#3 -0.0251 -0.1111 0.0609 False
RUN#1 RUN#2 -0.0012 -0.0872 0.0848 False
RUN#1 RUN#3 -0.0246 -0.1106 0.0614 False
RUN#2 RUN#3 -0.0233 -0.1093 0.0626 False

```

Figura 5: Risultato del TukeyHSD per le AP delle varie run

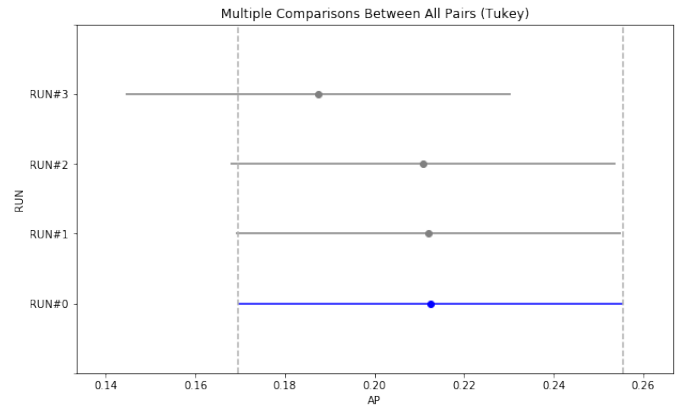


Figura 6: Plot del TukeyHSD test per le AP delle run

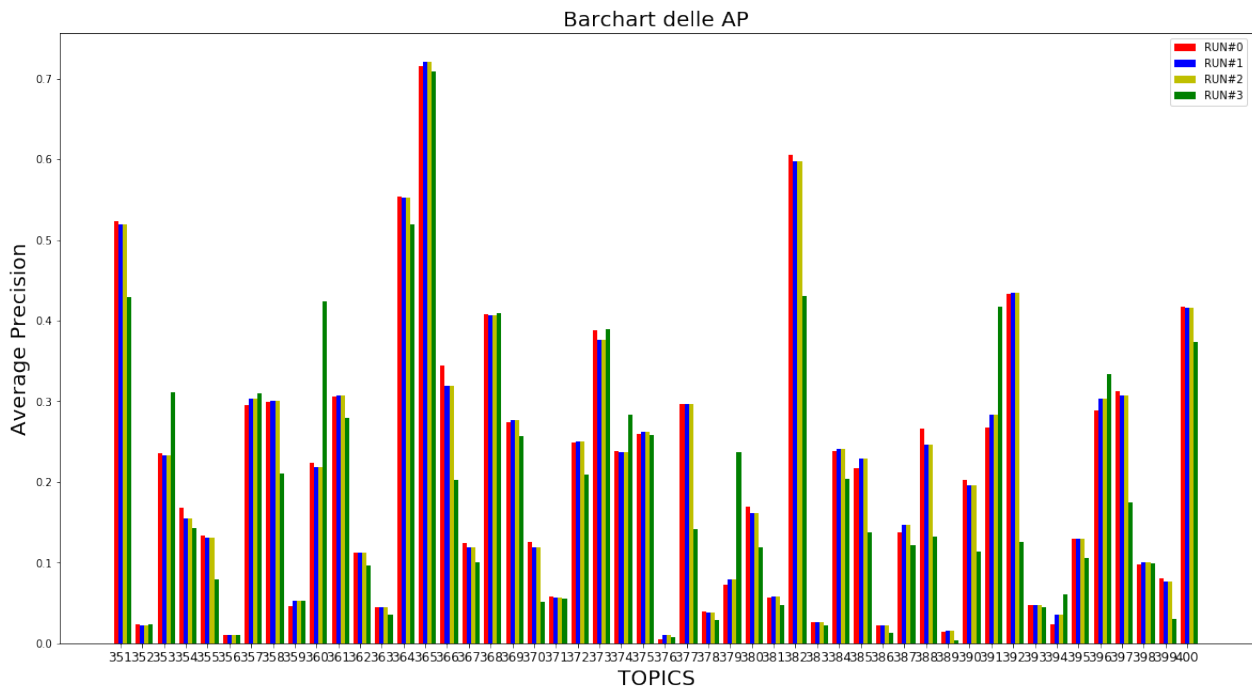


Figura 7: Ap per ogni topic, per ogni run