# Συστήματα Ανάκτησης – Φάση 4

# ΑΝΑΦΟΡΑ ΠΑΡΑΔΩΣΗΣ

(**Όνομα:** Αναστάσιος Παπαπαναγιώτου, **ΑΜ:** 3200143, **Email:** p3200143@aueb.gr)

(**Όνομα:** Φοίβος Παπαθανασίου, **ΑΜ:** 3200138, **Email:** p3200138@aueb.gr)

Αρχικά δημιουργούμε όλους τους συνδυασμούς για συγκεκριμένες τιμές παραμέτρων για τα similarities **BM25**, **LMJ** και μετά όλες τις δυάδες συνδυασμών που θα είναι ο συνδυασμός για το **MultiSimilarity**. Δημιουργούμε το **MultiSimilarity** για όλους τους συνδυασμούς με χρήση for και τους προσθέτουμε όλους σε μια λίστα.

Το υπόλοιπο κομμάτι παραμένει ίδιο με τις φάσεις 1, 2 με τη διαφορά ότι ο **IndexBuilder** παίρνει ως παράμετρο ένα similarity function. Επομένως, για κάθε MultiSimilarity τρέχουμε τον κώδικα της φάσης 1/2, και δημιουργούμε ένα αρχείο με τα αποτελέσματα.

Τα αρχεία αυτά επειδή είναι πολλά σε πλήθος δημιουργήσαμε ένα bash script για να γίνει πιο γρήγορα το οποίο εκτελεί το command του trec_eval για κάθε αρχείο. Για τους συνδυασμούς BM25+ClassicSimilarity, BM25+LMJ, LMJ+ClassicSimilarity δημιουργείται ένα αρχείο για καθένα που περιέχει τα αποτελέσματα του trec_eval.

Τοποθετούμε στο ίδιο directory όπου κάναμε extract το trec_eval.zip τα δύο αρχεία qrels.txt, όλα τα αρχεία για τους συνδυασμούς των **MultiSimilarity**, και το bash script. Τρέχουμε το bash script και στη συνέχεια βλέπουμε τα αρχεία που δημιουργήθηκαν από αυτό.

### Ανάλυση Παραμέτρων

Καθώς το σύνολο των συνδυασμών είναι μεγάλο, δημιουργήσαμε ένα python script που διαβάζει τα αποτελέσματα από το trec_eval και κρατάει όλους τους συνδυασμούς που βρίσκονται στο X-th percentile. Συγκεκριμένα, για κάποιο (ορίζεται ως CLI argument) από τα αρχεία με τα αποτελέσματα του script, διαβάζει τα αποτελέσματα και αποθηκεύει σε μια λίστα το αποτέλεσμα για όλα τα queries (all). Μετά για τα αρχεία του precision και map υπολογίζει το X-th percentile για το precision (το X ορίζεται ως CLI argument), το X-th percentile για το map και βρίσκει ποιοι συνδυασμοί βρίσκονται στο X-th percentile. Τέλος βρίσκει το intersection μεταξύ των συνδυασμών του map και του prec. Το X θέλουμε να είναι ο μεγαλύτερος αριθμός τέτοιος ώστε το intersection να είναι μη κενό σύνολο.

Οι καλύτερες τιμές παραμέτρων που βρήκαμε είναι (95-th percentile):

- **LMJ+BM25:** k1=3.0, b=0.25, λ=0.9 ή k1=3.0, b=0.5, λ=0.9
- **LMJ+Classic:** λ=0.9
- **BM25+Classic:** k1=3.0, b=0.25

Επομένως οι καλύτερες τιμές είναι k1=3.0, b=0.25, λ=0.9. Στην προηγούμενη φάση βρήκαμε πως το λ=0.9 ήταν το καλύτερο ενώ το k1=3, b=0.25 ήταν τα καλύτερα (ίδια με αυτή τη φάση).

- LMJ+Classic με λ=0.9:

| Q | MAP | Precision@5 | Precision@10 | Precision@15 | Precision@20 |
|---|---|---|---|---|---|
| Q01 | 0.7467 | 0.8 | 0.9 | 0.8667 | 0.65 |
| Q02 | 0.2852 | 0.6 | 0.4 | 0.2667 | 0.2 |
| Q03 | 0.6199 | 1.0 | 0.6 | 0.4667 | 0.4 |
| Q04 | 0.0667 | 0.0 | 0.3 | 0.2 | 0.15 |
| Q05 | 0.2691 | 0.4 | 0.3 | 0.3333 | 0.35 |
| Q06 | 0.0613 | 0.0 | 0.2 | 0.2 | 0.2 |
| Q07 | 0.2134 | 0.2 | 0.2 | 0.2 | 0.2 |
| Q08 | 0.8661 | 1.0 | 1 | 0.8 | 0.6 |
| Q09 | 0.2103 | 0.6 | 0.4 | 0.2667 | 0.2 |
| Q10 | 0.2711 | 0.6 | 0.4 | 0.2667 | 0.2 |
| all | 0.3610 | 0.52 | 0.47 | 0.3867 | 0.3150 |

- BM25+Classic με b=0.25, k1=3.0

| Q | MAP | Precision@5 | Precision@10 | Precision@15 | Precision@20 |
|---|---|---|---|---|---|
| Q01 | 0.7779 | 0.8 | 0.8 | 0.8 | 0.7 |
| Q02 | 0.3092 | 0.6 | 0.3 | 0.2667 | 0.25 |
| Q03 | 0.6200 | 0.8 | 0.6 | 0.5333 | 0.4 |
| Q04 | 0.0490 | 0.0 | 0.1 | 0.2 | 0.15 |
| Q05 | 0.2746 | 0.4 | 0.3 | 0.3333 | 0.35 |
| Q06 | 0.0985 | 0.2 | 0.2 | 0.2 | 0.2 |
| Q07 | 0.2527 | 0.2 | 0.2 | 0.3333 | 0.3 |
| Q08 | 0.8892 | 1.0 | 1 | 0.8 | 0.6 |
| Q09 | 0.1989 | 0.6 | 0.4 | 0.2667 | 0.25 |
| Q10 | 0.2917 | 0.6 | 0.4 | 0.2667 | 0.2 |
| all | 0.3762 | 0.52 | 0.43 | 0.4 | 0.34 |

- BM25+LMJ με b=0.25, k1=3.0, λ=0.9

| Q | MAP | Precision@5 | Precision@10 | Precision@15 | Precision@20 |
|---|---|---|---|---|---|
| Q01 | 0.7831 | 0.8 | 0.9 | 0.8 | 0.7 |
| Q02 | 0.3087 | 0.6 | 0.3 | 0.2667 | 0.25 |
| Q03 | 0.6150 | 0.8 | 0.6 | 0.5333 | 0.4 |
| Q04 | 0.0521 | 0.0 | 0.2 | 0.2 | 0.15 |
| Q05 | 0.2622 | 0.4 | 0.3 | 0.2667 | 0.35 |
| Q06 | 0.0963 | 0.2 | 0.3 | 0.2 | 0.2 |
| Q07 | 0.2401 | 0.2 | 0.2 | 0.1333 | 0.3 |
| Q08 | 0.8826 | 1.0 | 1 | 0.8 | 0.6 |
| Q09 | 0.2202 | 0.6 | 0.4 | 0.2667 | 0.25 |
| Q10 | 0.2917 | 0.6 | 0.4 | 0.2667 | 0.2 |
| all | 0.3752 | 0.52 | 0.46 | 0.3733 | 0.34 |

Παρατηρούμε ότι οι διαφορές με τα αποτελέσματα της δεύτερης φάσεις είναι πολύ μικρά, δηλαδή υπήρξε μια πολύ μικρή αύξηση στα αποτελέσματα με τη χρήση MultiSimilarity.

**Πηγές:** eclass, Lucene's documentation

**Directory map:** τα αρχεία analysis.py που έγινε η ανάλυση και το bash script βρίσκονται στο directory: "src/main/resources/output"