Συστήματα ανάκτησης πληροφοριών

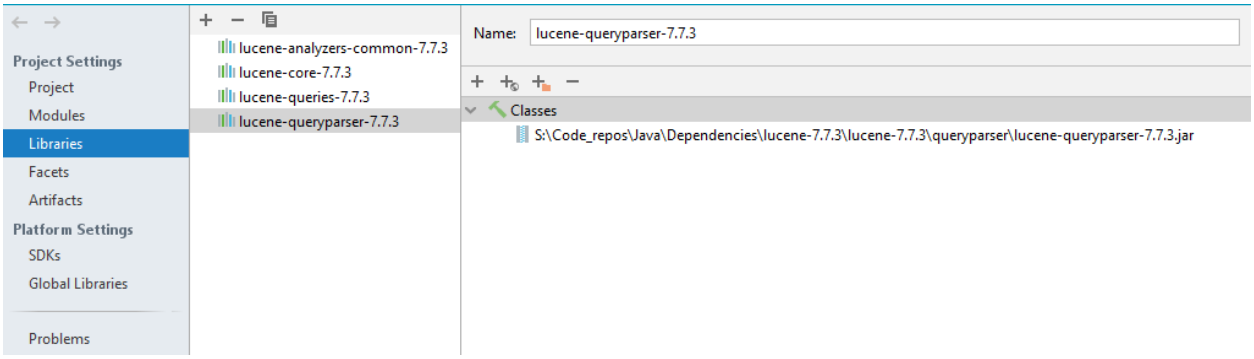
Προγραμματιστική εργασία, φάση 1

Δημήτρης Μπάστας, 3130139

Φίλιππος Δουραχαλής, 3170045

Η εκπόνηση της εργασίας πραγματοποιήθηκε στο περιβάλλον Intellij IDEA.

Αρχικά φορτώνουμε τις απαραίτητες βιβλιοθήκες της Lucene μέσα από το περιβάλλον του Intellij επιλέγοντας File > Project Structure > Libraries > + (New Project Library) > Java και στη συνέχεια προσθέτοντας τα αρχεία jar που φαίνονται στην εικόνα:



Κατά την πρώτη εκτέλεση του κώδικα απαιτείται να δώσουμε 3 arguments:

1. Την τοποθεσία της συλλογής αρχείων
2. Το directory στο οποίο θα αποθηκευτεί το directory και
3. Την τοποθεσία του αρχείου που περιέχει τα ερωτήματα (queries.text)

Επίσης μπορούμε να τροποποιήσουμε την μεταβλητή της κλάσης TREC_EVAL_PATH ώστε να δείχνει στον φάκελο που περιέχει το trec_eval.exe, ώστε το αρχείο των αποτελεσμάτων και το μορφοποιημένο αρχείο qrels.text να δημιουργηθούν απευθείας εκεί και να μην χρειαστεί να τα μετακινήσουμε (π.χ. “C:/Users/p3170045/Downloads/trec_eval”).

Τέλος δίνουμε στην μεταβλητή QUERIES_PATH της DocRetreival την τοποθεσία που έχουμε το αρχείο qrels.text.

Για την υλοποίηση της εργασίας δημιουργήσαμε τις εξής 5 διακριτές κλάσεις:

DocRetrieval: Πρόκειται για την κλάση που περιέχει την μέθοδο main. Μέσα σε αυτή διαβάζουμε τα απαραίτητα αρχεία και καλούμε τις μεθόδους των υπόλοιπων κλάσεων για να πάρουμε το ευρετήριο και τα ερωτήματα αναζήτησης του αρχείου queries.text. Στη συνέχεια κάνουμε parse τα ερωτήματα που επιστράφηκαν και παράγουμε το αρχείο results.text με τα αποτελέσματα.

Indexer: Η κλάση αυτή είναι υπεύθυνη για την δημιουργία των Documents και των πεδίων τους ώστε να προστεθούν στο Index.

DocParser: Η μέθοδος parse(String file) της κλάσης αυτής μας επιτρέπει να διαβάσουμε τα κείμενα της συλλογής ως ενιαία συμβολοσειρά και να την επεξεργαστούμε κατάλληλα για να πάρουμε τα πεδία κάθε κειμένου ξεχωριστά.

RawQueryParser: Αντίστοιχα η μέθοδος parse(String file) της συγκεκριμένης κλάσης δουλεύει με παρόμοιο τρόπο, μετατρέποντας το αρχείο που δίνεται σε συμβολοσειρά την οποία επεξεργάζεται για να πάρει τα πεδία id (.I), description (.W), authors (.A) και keywords (.N) του κάθε ερωτήματος και να τα συνδυάσει ώστε να πάρει ένα ολοκληρωμένο ερώτημα

MyDoc: Η κλάση αυτή αναπαριστά ένα άρθρο από τη συλλογή κειμένων και αποθηκεύει τις τιμές όλων των πεδίων που μπορεί να συναντήσουμε σε ένα κείμενο.

Για την προεπεξεργασία των κείμενων για να είναι σε κατάλληλη μορφή ώστε να μπορεί να τα διαχειριστεί ο Indexer χρησιμοποιήσαμε την εντολή str.split("pattern") με το κατάλληλο regex ώστε να πάρουμε το κείμενο μεταξύ δυο tags της μορφής *.A*, *.T* κ.τ.λ. Επίσης κρατάμε τα tags ώστε να συμπληρώσουμε κατάλληλα τα fields σε κάθε document που θα δημιουργήσουμε χωρίς να μας δημιουργούν πρόβλημα τα κενά fields. Έπειτα δημιουργούμε ένα αντικείμενο MyDoc για κάθε κείμενο της συλλογής, προσθέτοντας τις τιμές των fields, και το βάζουμε σε μια λίστα, την οποία επιστρέφουμε στην κλάση DocRetrieval.

Μέσω αυτής ανοίγουμε τον αρχείο όπου θα αποθηκευτεί το Index και χρησιμοποιώντας τον Indexer δημιουργούμε για κάθε document που επιστράφηκε ένα αντικείμενο τύπου Document της Lucene με το μοναδικό id ως StringField καθώς μας ενδιαφέρει να είναι δυνατή η αναζήτηση και η ανάκτηση (Field.Store.YES) βάσει αυτού. Αποθηκεύουμε τον τίτλο ως StoredField καθώς έχει μικρό μέγεθος ώστε να ανακτηθεί μαζί με το id των κειμένων, ενώ τέλος κάνουμε concatenate όλα τα άλλα χρήσιμα πεδία που χρησιμοποιούνται στα queries (τίτλος, abstract, συγγραφείς, ημερομηνία και keywords) και τα βάζουμε σε ένα TextField έτσι ώστε να γίνουν tokenized και indexed.

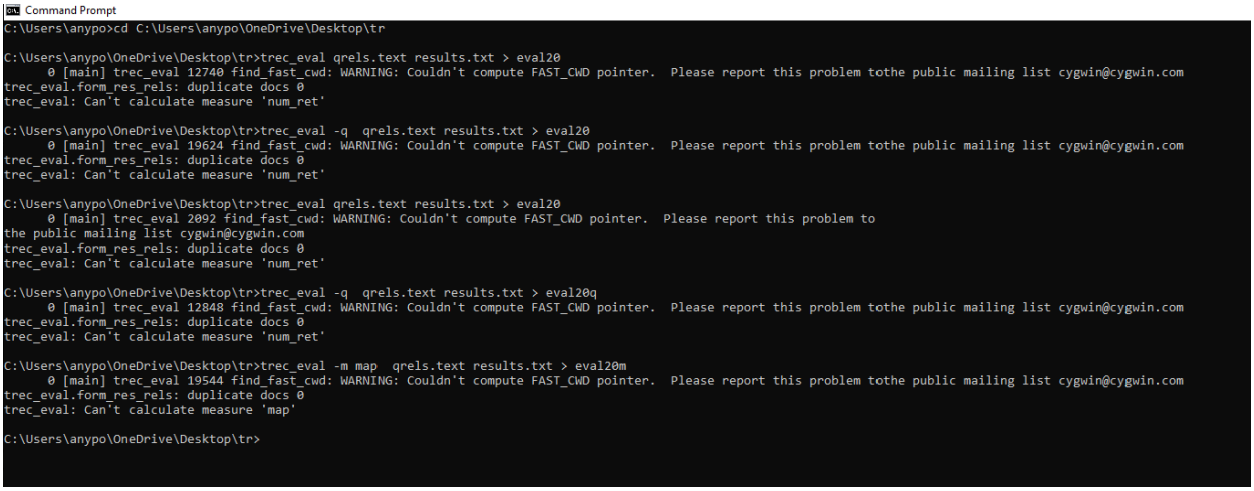
Αφού ολοκληρώσουμε τη δημιουργία του index τροποποιούμε το αρχείο qrels.text της CACM ώστε να έχει την κατάλληλη μορφοποίηση για να χρησιμοποιηθεί στο trec_eval. Χρησιμοποιούμε τον EnglishAnalyzer για να γίνει η επεξεργασία των κειμένων και των ερωτημάτων σε συνδυασμό με το ClassicSimilarity() για το search. Αφού δημιουργήσουμε το Index, δημιουργούμε τον searcher και

κάνουμε με αντίστοιχο τρόπο parse το query.text. Τρέχουμε τον searcher με κάθε query και παίρνουμε μέσω της TopDocs τα N σημαντικότερα αποτελέσματα τα όποια γραφούμε με κατάλληλη μορφή στο results.txt. Τρέχουμε το πρόγραμμα αλλάζοντας την τιμή της μεταβλητής RESULTS της DocRetrieval για να προσδιορίσουμε κάθε φορά το πλήθος αποτελεσμάτων που πρέπει να καταγραφούν.

Χρησιμοποιώντας το trec_eval με την εντολή “trec_eval [-q, -m] qrels.text results.text > (αρχείο eval)” βρήκαμε τα Map και P_k ως ορίζονται στον κάτω πίνακα .

k	20	30	50
MAP	0.2520	0.2700	0.2866
P_5	0.4154	0.4154	0.4154
P_10	0.3615	0.3615	0.3615
P_15	0.3128	0.3128	0.3128
P_20	0.2721	0.2721	0.2721

Επιπλέον στοιχειά συμπεριλαμβάνονται στα αρχεία eval, eval_detailed και eval_map:



Φαίνεται ότι το μέσο precision ανεβαίνει όσο αυξάνεται το k αλλά όχι στα 20 πρώτα αρχεία (p_5 έως p_20).