Συστήματα Ανάκτησης Πληροφοριών 2021

1η προγραμματιστική εργασία – Μοντέλο Ανάκτησης Διανυσματικού Χώρου

Αιμιλία Δανοπούλου 3170033

Έλενα Μηνά 3170108

Η συλλογή κειμένων πάνω στην οποία εργαστήκαμε ήταν η LISA. Τα κείμενα της συλλογής βρίσκονται στα αρχεία LISA-Documents που περιέχει τα εξής: τα κείμενα χωρισμένα σε διάφορα αρχεία με όνομα LISAX.XXX όπου Χ κάποιος αριθμός.

Σημείωση: Τοποθετήσαμε σε διαφορετικούς φακέλους τα κείμενα καθώς και τα LISAQUE και LISARJ.NUM.

1.Προεπεξεργασία:

Με σκοπό να μπορέσουμε να διαβάσουμε τα κείμενα LISA , δημιουργήσαμε την κλάση TXTParsing. Η κλάση αυτή παίρνει σαν όρισμα ένα αρχείο σε μορφή String κάθε φορα που καλείται. Σύμφωνα με την δομή που είχαν τα κείμενα , κάναμε split το εκάστοτε αρχείο με βάση την συμβολοσειρά «\*\*\*\*\*\*\*\*…\*\*\r\n» +++

# 2. Δημιουργία Ευρετηρίου

Για την δημιουργία του ευρετηρίου δημιουργήσαμε την κλάση CreateIndex. Σε αυτή την κλάση ορίσαμε την μέθοδο readFolderContents που επιστρέφει ένα ArrayList<String> με όλα τα ονόματα των αρχείων στον σχετικό φάκελο, την οποία και καλούμε στην μέθοδο/constructor CreateIndex(). Επιλέξαμε τον English Analyzer και συνάρτηση ομοιότητας την ClassicSimilarity τα οποία περάσαμε σαν παραμέτρους σε ένα αντικείμενο τύπου IndexWriterConfig της Lucene και έπειτα για κάθε αρχείο καλέσαμε την μέθοδο parse της κλάσης ΤΧΤparsing η οποία επιστρέφει ένα ArrayList με αντικείμενα MyDoc, όπου το κάθε αντικείμενο είναι ένα κείμενο στο συγκεκριμένο αρχείο το ποιο γράφουμε στο index μας.

3. Query Parsing και αναζήτηση στο ευρετήριο

Για τα queries που μας έδινε η συλλογή LISA δημιουργήσαμε την κλάση QueryParsing η οποία επιστρέφει μια λίστα με αντικείμενα MyQuery για να κρατάμε τα queries μας. Στην κλάση αυτή κάνουμε split το περιεχόμενου του αρχείου LISA.QUE στον χαρακτήρα «#». Έπειτα αφού έχουμε ανακτήσει το id και το body του κάθε query τα περνάμε σαν παραμέτρους σε ένα νέο αντικείμενο MyQuery και τα προσθέτουμε στην λίστα parsed\_queries σε κάθε iteration και τέλος επιστρέφουμε την λίστα.

# 4.Searcher

Στην κλάση Searcher κάνουμε την αναζήτηση για κάθε query πάνω στο ευρετήριο που δημιουργήθηκε. Η αναζήτηση γίνεται πάνω στο πεδίο contents που περιέχει το title και το body των κειμένων και δημιουργούμε αρχεία με τα 20, 30 και 50 πρώτα ανακτηθέντα κείμενα που έχουν αντίστοιχα ονομασία our\_results\_k=20, our\_results\_k=30, our\_results\_k=50 καθώς και ένα αρχείο με όλα τα κείμενα που επιστρέφονται με όνομα our\_results.

# 5.LISARJ.NUM parsing

Για το δίαβασμα του αρχείου LISARJ.NUM το οποίο περιέχει τις σωστές απαντήσεις για τα queries της συλλογής, δημιουργήσαμε την κλάση RelevantAnswersParser. Στην μέθοδο parse\_relevant() φτιάχνουμε ένα αρχείο με όνομα rel\_results.test και κάνουμε parse το αρχείο με τις σωστές απαντήσεις ώστε να είναι σε μορφή αναγνωρίσιμη από το εργαλείο trec\_eval.

# 6.Trec Eval

Έχοντας πλέον δημιουργήσει τα αρχεία με τα αποτελέσματα και σε κατάλληλη μορφή για το trec\_eval, καθώς και την δημιουργία του rel\_results με τα σωστά αποτελέσματα που μας παρέχει η συλλογή LISA μπορούμε να κάνουμε τους ελέγχους.

1.trec\_evail rel\_results.test our\_results\_k=20.test

A screenshot of a video game

Description automatically generated with medium confidence

2.trec\_evail rel\_results.test our\_results\_k=30.test

A screenshot of a video game

Description automatically generated with medium confidence

3.trec\_evail rel\_results.test our\_results\_k=50.test

A picture containing text

Description automatically generated

Το precision για 5,10,15 και 20 είναι το ίδιο και για τα 3 αρχεία

A screenshot of a video game

Description automatically generated with low confidence

|  |  |  |  |
| --- | --- | --- | --- |
|  | K = 20 | K = 30 | K = 50 |
| map | 0.2036 | 0.2184 | 0.2242 |
| P\_5 | 0.2800 | 0.2800 | 0.2800 |
| P\_10 | 0.1750 | 0.1750 | 0.1750 |
| P\_15 | 0.1667 | 0.1667 | 0.1667 |
| P\_20 | 0.1450 | 0.1450 | 0.1450 |
|  |  |  |  |