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. Στις παρακάτω ενότητες περιγράφονται τα βήματα της επεξεργασίας των αρχείων της συλλογής αυτής, καθώς και οι κλάσεις που χρησιμοποιήθηκαν προκειμένου να εκτελεστεί η διαδικασία της ανάκτησης κειμένων με βάση τα δοθέντα ερωτήματα.

1.Προεπεξεργασία:

Για την λειτουργία της εφαρμογής ήταν απαραίτητη η προεπεξεργασία τριών τύπων αρχείων:

- Τα αρχεία με όνομα LISAX.XXX στα οποία περιλαμβάνονται τα κείμενα της συλλογής,
- Το αρχείο LISA.QUE τα οποίο περιλαμβάνει διάφορα ερωτήματα προς την συλλογή,
- Το αρχείο LISARJ.NUM το οποίο χρησιμοποιήθηκε για να επαληθεύσουμε την εγκυρότητα τον απαντήσεων που επιστρέφει η εφαρμογή.

Προκειμένου να μπορέσουμε να διαβάσουμε τα κείμενα LISA, δημιουργήσαμε την κλάση TXTParsing. Η κλάση αυτή περιέχει μια static μέθοδο η οποία παίρνει σαν όρισμα το όνομα ενός αρχείου σε μορφή String.

Η μέθοδος αυτή διαβάζει το αρχείο που δόθηκε ως όρισμα και το μετατρέπει σε μια ενιαία συμβολοσειρά μέσω της μεθόδου IO.ReadEntireFileIntoAString(String filename). Σύμφωνα με την δομή των αρχείων LISAX.XXX κάναμε split την παραπάνω συμβολοσειρά με βάση το μοτίβο «********…**\r\n», ξεχωρίζοντας έτσι το κάθε κείμενο της συλλογής και αποθηκεύοντας το σε μία μεταβλητή τύπου String, εντός ενός array.

Έπειτα, επεξεργαστήκαμε ξεχωριστά κάθε μεταβλητή τύπου String που αντιστοιχεί σε ένα κείμενο προκειμένου να προσδιορίσουμε το id, τον τίτλο και το περιεχόμενο του κειμένου.
Τα τρία αυτά πεδία εντοπίζονται εντός του String παίρνοντας τα κατάλληλα substrings και κάνοντας κατάλληλους ελέγχους. Συγκεκριμένα, ως όρια των substrings χρησιμοποιούνται οι θέσεις των χαρακτήρων «\r\n» και «.», καθώς και ο έλεγχος isWhitespace(char c) προκειμένου να εντοπιστεί η θέση του πρώτου χαρακτήρα που δεν είναι whitespace, ο οποίος σηματοδοτεί την έναρξη του body.
Εφόσον εντοπιστούν τα περιεχόμενα των πεδίων δημιουργείται ένα αντικείμενο τύπου MyDoc, με πεδία id, title και body, το οποίο αντιπροσωπεύει το εκάστοτε κείμενο της συλλογής που επεξεργάστηκε.

Προκειμένου να μπορέσουμε να απαντήσουμε στα queries που αφορούν την συλλογή υλοποιήσαμε την κλάση QueryParsing. Η κλάση αυτή διαβάζει και επεξεργάζεται το αρχείο

LISA.QUE. Όμοια με την κλάση TXTParsing, η κλάση αυτή χρησιμοποιεί την μέθοδο IO.ReadEntireFileIntoAString(String filename) και διαχωρίζει την ενιαία συμβολοσειρά που επιστρέφεται με βάση τον χαρακτήρα «#». Στη συνεχεία, αφαιρούνται από την συμβολοσειρά που αντιστοιχεί σε κάθε query τυχόν whitespace characters στην αρχή και το τέλος αυτής, μέσω της μεθόδου trim(). Έπειτα, προσδιορίζονται τα τμήματα της συμβολοσειράς που αντιστοιχούν στο id και στο body του query. Αυτό γίνεται παίρνοντας substrings σε κατάλληλα σημεία, με βάση την θέση του χαρακτήρα «\n». Στο τέλος της διαδικασίας αυτής δημιουργείται ένα αντικείμενο τύπου MyQuery, με πεδία ID και body, το οποίο αντιπροσωπεύει το εκάστοτε query που επεξεργάστηκε.

Τέλος, προκειμένου να ελέγξουμε την ορθότητα της λειτουργίας της εφαρμογής ήταν απαραίτητο να φέρουμε σε κατάλληλη μορφή το αρχείο LISARJ.NUM. Η διαδικασία αυτή συμβαίνει μέσω της κλάσης RelevantAnswersParser. Η μέθοδος parse_relevant(…) της κλάσης αρχικά δημιουργεί ένα νέο .txt αρχείο στο οποίο γράφει τα περιεχόμενα του LISARJ.NUM με κατάλληλη δομή. Και αυτή η μέθοδος κάνει χρήση της IO.ReadEntireFileIntoAString(String filename). Πάνω στην ενιαία συμβολοσειρά που επιστρέφεται εφαρμόζεται trim() και αντικατάσταση όλων των whitespace characters με τον χαρακτήρα «-». Στη συνέχεια, η συμβολοσειρά γίνεται split(…) με βάση τον χαρακτήρα «-». Εφόσον γνωρίζουμε πως οι αριθμοί που εμφανίζονται στο αρχείο αντιπροσωπεύουν πάντα με την ίδια σειρά τις πληροφορίες:

(query ID,αριθμός των σχετικών κειμένων, ID’s των κειμένων αυτών)

Με την χρήση κατάλληλων iteration πάνω στην συμβολοσειρά μπορούμε να εντοπίσουμε τις θέσεις των πληροφοριών αυτών και έπειτα να τις γράψουμε ανά γραμμή στο αρχείο “rel_results.test ”.

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 TXTparsing 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, το title 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 διάβασμα του αρχείου LISARJ.NUM το οποίο περιέχει τις σωστές απαντήσεις για τα queries της συλλογής, δημιουργήσαμε την κλάση RelevantAnswersParser. Περισσότερες λεπτομέρειες για την μέθοδο parse_relevant της κλάσης αυτής παρουσιάζονται στην ενότητα 1.

6.MainApp

Για την εκτέλεση της εφαρμογή χρησιμοποιείται η κλάση MainApp.java. Εκεί, ο χρήστης καλείται να δώσει ως εισόδους τα εξής:

- Το path προς τον φάκελο όπου βρίσκονται τα αρχεία της συλλογής (LISAX.XXX)
- Το path προς τον φάκελο όπου θέλει να γίνει η δημιουργία του ευρετηρίου
- Το path προς το αρχείο LISA.QUE
- Το path προς το αρχείο LISARJ.NUM

Σημείωση: η εφαρμογή είναι δομημένη να λειτουργεί τοποθετώντας τα αρχεία LISAX.XXX, LISA.QUE, LISARJ.NUM *σε διαφορετικούς φακέλους.*

7.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

2.trec_evail rel_results.test our_results_k=30.test

```
map                     all     0.2184
gm_map                  all     0.0140
Rprec                   all     0.2546
bpref                   all     0.3490
```

3.trec_evail rel_results.test our_results_k=50.test

```
map                     all     0.2242
gm_map                  all     0.0218
Rprec                   all     0.2546
bpref                   all     0.4174
```

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

```
P_5                     all     0.2800
P_10                    all     0.1750
P_15                    all     0.1667
P_20                    all     0.1450
```

	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