Ανάκτηση Πληροφορίας

Γκόγκος Κώστας Α.Μ 1611

Ζούμπος Μιχάλης Α.Μ 1484

Ζουμπαλιδης Έντγκαρ ΑΜ 1483

1.Ανάλυση κειμένων & Κατασκευή κειμένων

Παίρνουμε τα τέσσερα αρχεία yelp(business , checkin , review , user) σε μορφή .txt. Κάνοντας open , δημιουργούνται τέσσερα αντίστοιχα κανονικοποιημένα αρχεία .txt με τα οποία και δουλεύουμε . Αυτό γίνετε στην κλάση project στο κουμπί open. Εκεί δημιουργούμε 4 bufferedreaders τα οποία διαβάζουν τα 4 μη κανονικοποιημένα αρχεία και τα “καθαρίζουμε” από τα διάφορα σύμβολα, τα οποία βρίσκονται στο αρχείο punctuation.txt . Τα αποθηκεύουμε στους πίνακες τύπου string στις κλάσεις Users,business,checkIn και review αντιστοίχως. Στην συνέχεια δημιουργούμε τα αντίστοιχα κανονικοποιημένα αρχεία εισόδου μέσω των πινάκων.

2.Κατασκευή ευρετηρίου

Τα ευρετήρια κατασκευάζονται στην κλάση directory . Αρχικά καλούμε την συνάρτηση StandardAnalyzer() της Lucene για να φτιάξουμε τον αναλυτή μας . Στην συνέχεια δίνουμε το path για την τοποθεσία όπου θα δημιουργηθούν τα ευρετήρια και καλούμε την συνάρτηση Directory με όρισμα το path. Μετά δίνουμε το path όπου βρίσκονται τα κανονικοποιημένα αρχεία μας και καλούμε την συνάρτηση indexFileOrDirectory με όρισμα αυτό το path.Έτσι ώστε να προστεθούν τα αρχεία εισόδου στην ουρά έτσι ώστε να δημιουργηθούν τα ευρετήρια.

Χρησιμοποιείται η μέθοδος doc.add() η οποία διαβάζει τα tokens και το path του κάθε αρχείου εισόδου.
Μετά χρησιμοποιείται η μέθοδος writer.addDocument() η οποία δίνει τα αρχεία στον writer.
Επίσης χρησιμοποιείται η μέθοδος FSDirectory.open() η οποία επιστρέφει το path των αρχείων εισόδου για να κληθούν οι παραπάνω 2 μέθοδοι.

3 .Επεξεργασία ερώτησης

Πατώντας το Searchκαλούμε την μέθοδο Search()της κλάσης Projectμε όρισμα τον/τους όρους της ερώτησης. Αρχικά καλούμε τις μεθόδους DirectoryReader.open() για να πάρουμε τα ευρετήρια,IndexSearcher() για να ψάξουμε τα ευρετήρια
,TopScoreDocCollector.create() για να πάρουμε τα αποτελέσματα της αναζήτησης.
Δίνουμε την ερώτηση και την αποθηκεύουμε σε μια μεταβλητή την οποία την διαβάζουμε μέσω της μεθόδου QueryParser(). Για να γίνει η ερώτηση εγγύτητας η είσοδος διασπάτε με τον κενό χαρακτήρα και αποθηκεύεται σε ένα πίνακα.
Για να γίνει ο ορθογραφικός έλεγχος φτιάξαμε μια φορά το αρχείο suggest.txt το οποίο περιέχει όλες τις λέξεις ,χωρίς διπλότυπα, που υπάρχουν και στα 4 αρχεία εισόδου μας.

4.Εκτέλεσης της ερώτησης

Καταρχήν ο χρήστηςμπορεί να αναζητήσει μέχρι 6 λέξεις και η λειτουργία διάταξηςμε βάση την εγγύτητα από 2 μέχρι 6 λέξεις.
Μέσω searcher.search() γίνετε η αναζήτηση της/των εισόδων και με την Directory.collector.topDocs().scoreDocs παίρνουμε την βαθμολογία μέσω της οποίας γίνεται η διάταξη των εγγράφων , η βαθμολογία αποθηκεύεται στον πίνακα ScoreDoc[] hits .
Αν ο πίνακας είναι κενός που σημαίνει ότι δεν βρέθηκε κάποιο αποτέλεσμα καλείτε ο ορθογραφικός έλεγχος .

Στον ορθογραφικό έλεγχο αν δεν έχει δοθεί ορθογραφικά ορθή ερώτηση , εμφανίζει κάποιες προτάσεις οι οποίες υπάρχουν στο αρχείο suggest.txt το οποίο περιγράψαμε πιο πάνω. Χρησιμοποιούνται οι κλάσεις RAMDirectory , SpellChecker που παίρνει σαν όρισμα το RAMdirectory , τον αναλυτή StandardAnalyzer ,την κλάση IndexWriterConfig που παίρνει σαν όρισμα τον αναλυτή μας, την κλάση InputStreamReader η οποία διαβάζει το αρχείο suggest.txt, την κλάση PlainTextDictionary που παίρνει όρισμα το inputStreamreader και τέλος την μέθοδο spellChecker.indexDictionary() για να μας επιστραφούν οι προτεινόμενες λέξεις για κάθε λέξη της εισόδου και εμφανίζονται τα αποτελέσματα σε περίπτωση που υπάρχουν. Σε περίπτωση που ο χρήστης θελήσει να δώσει ερώτηση εγγύτητας παίρνουμε την είσοδο (2-6 λέξεις) ψάχνουμε σε κάθε πίνακα των κλάσεων Users,business,checkIn και review στα οποία βρίσκονται τα αρχεία εισόδων για να βρεθούν οι αποστάσεις ανάμεσα στα επιμέρους στοιχεία της ερώτησης και εμφανίζουμε τα ονόματα των αρχείων σε αύξουσα σειρά.

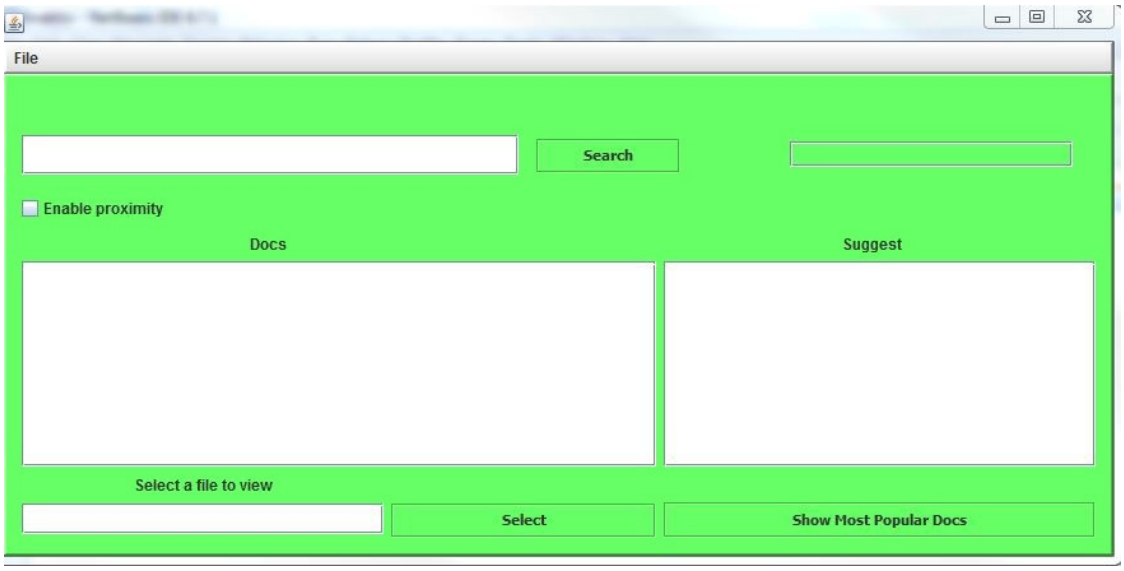
5.<u>Τονισμός των όρων της ερώτησης</u>

Πατώντας το κουμπί Selectκαλείται η μέθοδος openDocs() ,της κλάσης Project ,με όρισμα το όνομα του αρχείου που θέλουμε να εμφανίσουμε. Στην συγκεκριμένη μέθοδο γίνετε το παράθυρο BoltResult,της ομώνυμης κλάσης,visible . Εν συνεχεία καλείται η μέθοδος showDoc() , της κλάσης Project,με ορίσματα τον όρο της ερώτησης και το pathτου αρχείου .Στην μέθοδο showDoc() , αρχικά δημιουργούμε ένα Highlighter ,πάνω στο TextAreaBoltTexttου παραθύρου BoltResult .Όλο το κείμενο αποθηκεύεται σε ένα αλφαριθμητικό ,ψάχνουμε τους όρους της ερώτησης στο αλφαριθμητικό και μόλις βρίσκουμε κάποιον τον τονίζουμε με την εντολή h.addHighlight(point, point + length , DefaultHighlighter.DefaultPainter), όπου pointείναι η αρχή του όρου , point + length το τέλος

του όρου και , DefaultHighlighter.DefaultPainter ο χρωματισμός.

6.<u>Διάταξη ως προς την δημοτικότητα</u>

Όταν γίνετε η αναζήτηση (οποιαδήποτε από τις 2) , κρατάμε μετρητή για κάθε αρχείο οι οποίοι δείχνουν το πόσες φορές έχουν βρεθεί αποτελέσματα , σύμφωνα με τους μετρητές , γίνετε η διάταξη και η εμφάνιση των αρχείων στο TextAreaDocs.



7.<u>Γραφικό περιβάλλον και Προβολή των αποτελεσμάτων</u>

1.file(menu)

Open

Ο χρήστης Πατώντας open , αρχικά γίνεται διαγραφή των

ευρετηρίων από προηγούμενη χρήση , αν αυτά υπάρχουν.

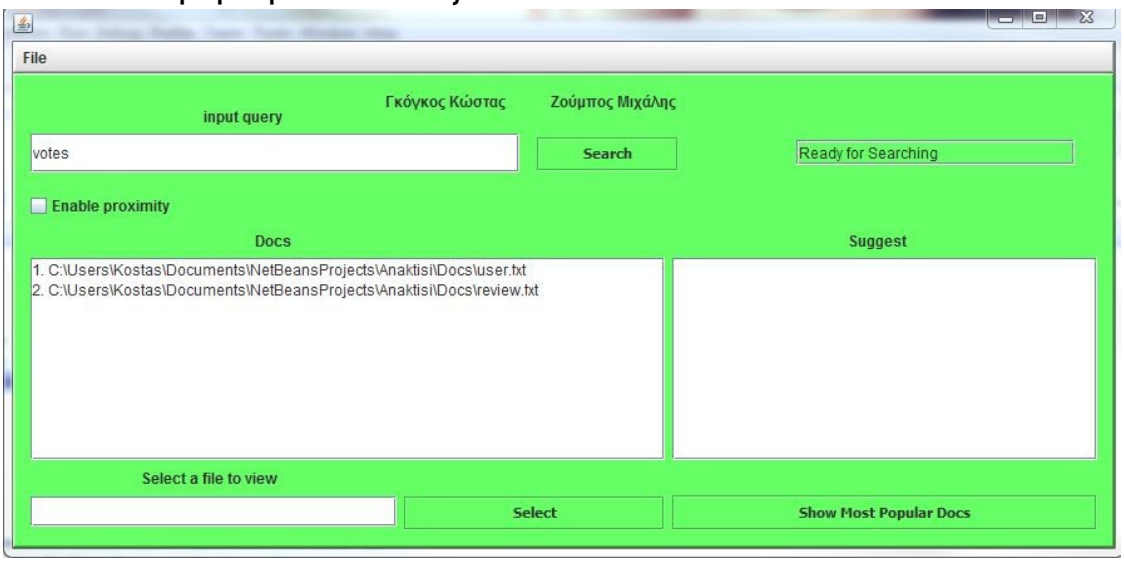
Εν συνεχεία γίνετε η κανονικοποίηση των αρχείων και η αποθήκευση τους στους πίνακες που έχουν περιγραφεί.

2.close

Ο χρήστης πατώντας closeτερματίζει την εφαρμογή.

2.Search

Ο χρήστης πατώντας το κουμπί searchκαι δίνοντας την ερώτηση στην μορφή που έχουμε περιγράψει παραπάνω ,πραγματοποιεί την αναζήτηση .Του δίνετε η επιλογή να δώσει ερώτημα εγγύτητας , τσεκάροντας το Enableproximity,αν δεν έχουν δοθεί από 2 μέχρι 6 λέξεις , εμφανίζεται μήνυμα λάθους. Σε περίπτωση που δεν έχει γίνει open , εμφανίζεται μήνυμα λάθους.



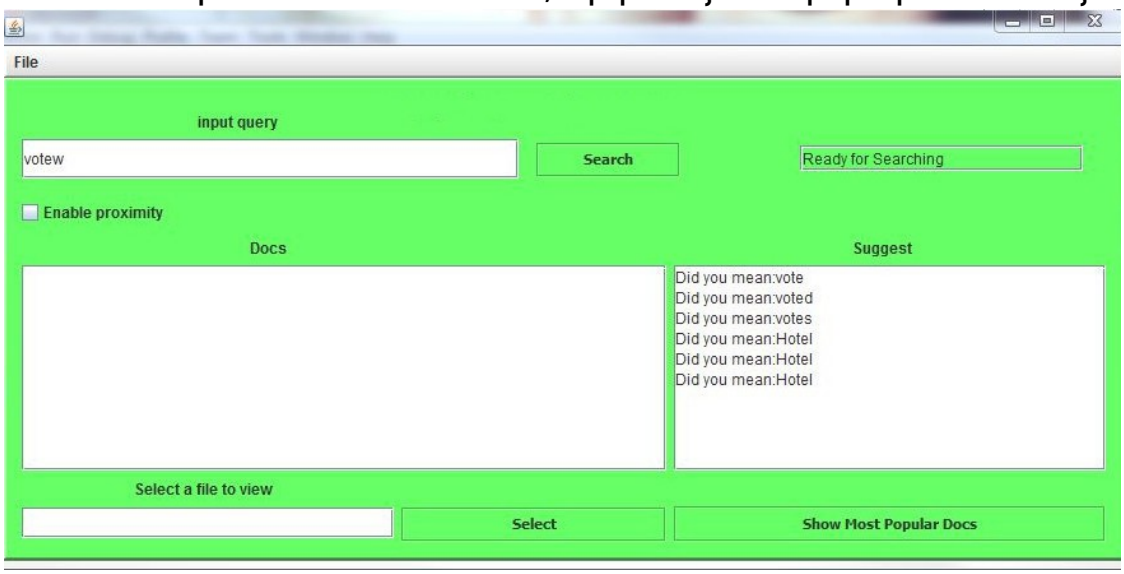
3.Docs

Είναι το TextArea στο οποίο εμφανίζονται τα αποτελέσματα των 2 ειδών αναζήτησης , ο χρήστης επιλέγοντας με το ποντίκι το αρχείο που θέλει να εμφανίσει , γίνετε αυτόματα η εμφάνιση του κειμένου που περιέχεται στο αρχείο με τονισμένους τους όρους της ερώτησης. Επίσης εμφανίζονται και τα αποτελέσματα της διάταξης των εγγράφων ως προς την δημοτικότητά τους. . Σε περίπτωση που ο χρήστης δεν έχει κάνει αναζήτηση και πατήσει μέσα στο TextArea ή διαλέξει αρχείο που δεν υπάρχει, εμφανίζεται μήνυμα λάθους.

4.Suggest

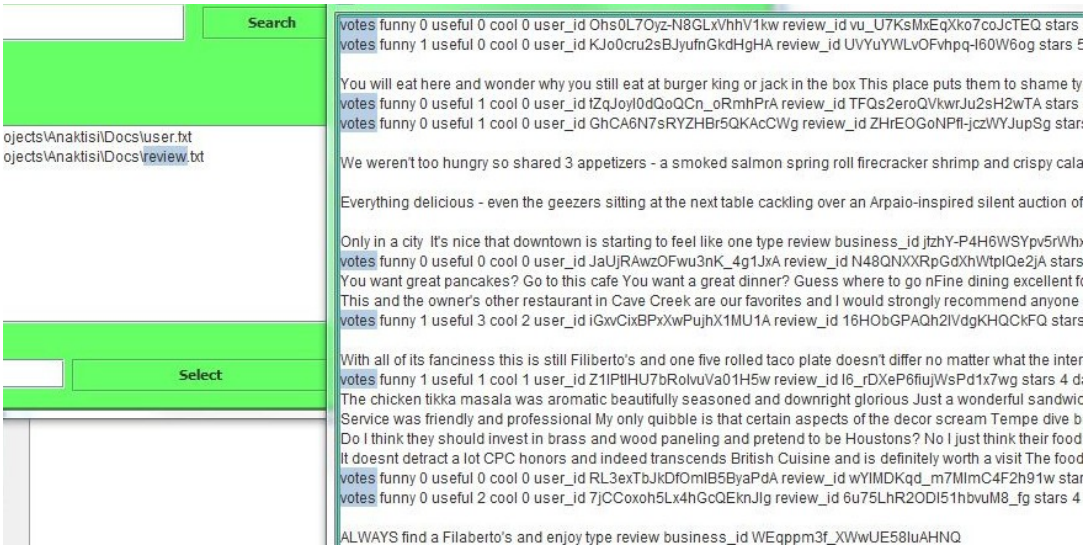
Είναι το TextArea στο οποίο εμφανίζονται οι προτεινόμενες λέξεις σε περίπτωση που η αναζήτηση δεν επιστρέψει αποτέλεσμα , επιλέγοντας με το ποντίκι την λέξη ,την οποία

θα θέλαμε να αναζητήσουμε ,γίνετε η αναζήτηση της λέξης. Σε περίπτωση που ο χρήστης δεν έχει κάνει αναζήτηση και πατήσει μέσα στο TextArea, εμφανίζεται μήνυμα λάθους.



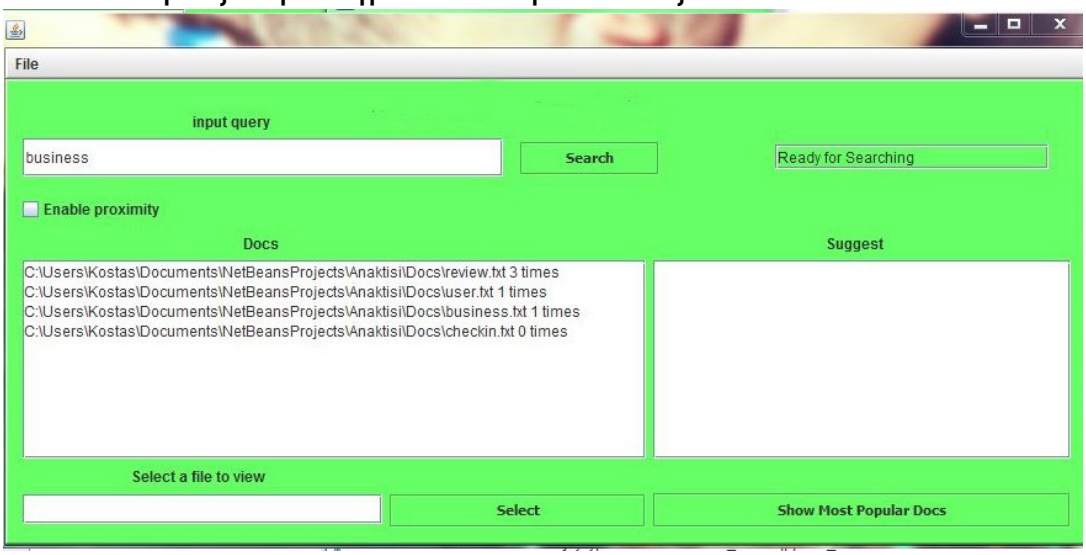
5.Select

Δίνοντας το όνομα του αρχείου στο TextFieldSelectafiletoView , εμφανίζεται το κείμενου που περιέχεται στο αρχείο με τονισμένους τους όρους της ερώτησης. Η εμφάνιση γίνετε σε ένα νέο παράθυρο (Boltresult). Σε περίπτωση που ο χρήστης δεν έχει κάνει αναζήτηση ή δεν έχει δώσει αρχείο εισόδου ,εμφανίζεται μήνυμα λάθους.



6.ShowMostPopularDocs

Πατώντας το κουμπί αυτό , εμφανίζονται και τα αποτελέσματα της διάταξης των εγγράφων ως προς την δημοτικότητά τους.



Οδηγίες εγκατάστασης:

Ο χρήστης πρέπει να έχει εγκατεστημένο το εργαλείο NetBeanskαι να κάνει importτις βιβλιοθήκες της lucene_48.
Τα αρχεία πρέπει να τα έχει σταθερά σε ένα directory.
Τα pathαλλάζουν χειροκίνητα στον κώδικα.
Όλεςοιενέργειεςγίνονταιαπότογραφικόπερι βάλλον.