Πήττας Νικόλαος, ΑΜ:5007

Ανάκτηση πληροφορίας
2η φάση εργασίας

GitHub repository link: https://GitHub.com/Nik-Pt/InformationRetrievalProject.git

Στόχος και λειτουργικότητα συστήματος:

Ο στόχος του συστήματος εξαρχής ήταν η εύκολη και γρήγορη αναζήτηση - εύρεση λέξεων ή φράσεων που εισάγει ο χρήστης σε διάφορα πεδία. Ως προς την λειτουργικότητα το σύστημα υποστηρίζει αναζήτηση λέξεων ή φράσεων τις οποίες εισάγει ο χρήστης καθώς και δυνατότητα επιλογής πεδίων που επιθυμεί να γίνει η αναζήτηση. Αφού ο χρήστης πατήσει αναζήτηση το σύστημα παρουσιάζει τα τραγούδια στα οποία βρέθηκε ο ορός αναζήτησης μαζί και με μια μικρή πρόταση στην οποία βρέθηκε για πρώτη φορά αυτός ο ορός. Τέλος δίνεται η δυνατότητα στον χρήστη να δει το ιστορικό των αναζητήσεων του αλλά και να ταξινομήσει τα αποτελέσματα με βάση το όνομα του καλλιτέχνη

Συλλογή:

Η συλλογή αποτελείται από έναν φάκελο με το όνομα csv ο οποίος περιέχει 19 .csv αρχεία. Συγκεκριμένα κάθε .csv αρχείο περιέχει έναν αριθμό τραγουδιών και έχει ως όνομα το όνομα του τραγουδιστή (πχ Το Eminem.csv περιέχει μόνο τραγούδια του Eminem). Η συγκεκριμένη συλλογή επιλέχθηκε διότι κατά την αναζήτηση ο χρήστης θα μπορεί ευκολά να καταλάβει σε ποιο αρχείο βρέθηκε ο ορός αναζήτησης καθώς το όνομα ενός τραγουδιστή ταυτίζεται με το όνομα του αρχείου στο οποίο ανήκει. Η συλλογή των αρχείων έγινε μέσα από το Kaggle , συγκεκριμένα από το παρακάτω link: https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset από το οποίο αφαίρεσα 2 αρχεία .csv λόγω προβλημάτων με τον κώδικα του συστήματος. Το κάθε αρχείο .csv όπως προαναφέρθηκε περιέχει έναν αριθμό τραγουδιών. Το κάθε τραγούδι έχει τα εξής πεδία: Αριθμός τραγουδιού (ο οποίος δεν χρησιμοποιείτε στην αναζήτηση), όνομα τραγουδιστή ,τίτλος τραγουδιού , όνομα άλμπουμ στο οποίο ανήκει το τραγούδι , έτος κυκλοφορίας , ακριβής ημερομηνία κυκλοφορίας και τα lyrics.
Αυτά τα πεδία είναι διαθέσιμα στον χρήστη για να τα χρησιμοποιήσει στην αναζήτηση.

Ανάλυση κειμένου και κατασκευή ευρετηρίου:

Για τα .csv αρχεία γίνεται μια προεπεξεργασία. Ειδικότερα γίνεται χρήση του Standard Analyzer που παρέχει η Lucene κάνοντας import το κατάλληλο αρχείο (import org.apache.lucene.analysis.standard.StandardAnalyzer) για απαλοιφή stop words αλλά και stemming το οποίο υποστηρίζει ο συγκεκριμένος Analyzer. Ιδανικά θα ήθελα να χρησιμοποιήσω τον Snowball analyzer ο οποίος παρέχει περισσότερες δυνατότητες σε σχέση με τον standard αλλά και τον Porter Stemmer για καλύτερο stemming ωστόσο καμία από τις δυο επιλογές δεν περιλαμβάνονταν στην έκδοση 9.5.0 της Lucene. Τα πεδία τα οποία χρησιμοποιώ για την δημιουργία του αρχείο αλλά και για τη αναζήτηση είναι: artist,title,album,year,date,lyrics. Για να υποστηρίζεται η αναζήτηση δημιουργώ έναν νέο φάκελο Index στον οποίο θα βρίσκεται το ευρετήριο.
Με την χρήση της κλάσης ReadCSV η οποία είναι υπεύθυνη για το διάβασμα κάθε αρχείου τύπου .csv δημιουργώ μια νέα λίστα ArrayList<Song> στην οποία αποθηκεύονται αντικείμενα τύπου Song (ένα αντικείμενο Song αντιστοιχεί σε ένα τραγούδι). Για κάθε τραγούδι στην λίστα δημιουργώ ένα νέο Document στο οποίο αποθηκεύω τα δεδομένα ενός τραγουδιού ανάλογα με το πεδίο τους. Τέλος με την χρήση του IndexWriter ο οποίος

επίσης παρέχεται από την Lucene αποθηκεύω κάθε Document στο ευρετήριο ώστε να τα χρησιμοποιήσω αργότερα στην αναζήτηση.

Αναζήτηση:

Ως προς τη αναζήτηση το σύστημα δίνει την δυνατότητα στον χρήστη να εισάγει την λέξη προς εύρεση αλλά και να επιλέξει το πεδίο στο οποίο θέλει να γίνει η αναζήτηση. Για την εκτέλεση της αναζήτησης γίνεται χρήση της MultiFieldQueryParser ώστε να μπορεί να γίνει σε παραπάνω από ένα πεδία. Πιο συγκεκριμένα χρησιμοποιώ 2 MultiFieldQueryParse ώστε να μπορώ να αναζητήσω και να εμφανίσω στα αποτελέσματα τραγούδια που περιέχουν τον ορό που εισήγαγε ο χρήστης αυτούσιο (αν υπάρχει) και στη συνέχεια τραγούδια που περιέχουν λέξεις οι οποίες προέρχονται από τον ορό αναζήτησης. Για να μπορώ να έχω πρόσβαση και στα 2 αποτελέσματα χρησιμοποιώ ένα BooleanQuery ώστε και κάνω parse 2 διαφορετικά query οπού το ένα θα περιέχει αυτούσιο τον ορό αναζήτησης και το δεύτερο θα χρησιμοποιηθεί για την εύρεση παραγόμενων λέξεων από τον αρχικό ορό αναζήτησης. Εκτός από μια απλή λέξη ο χρήστης μπορεί να εισάγει μια φράση η οποία μπορεί πέρα από λέξεις ή γράμματα να περιέχει επίσης και αριθμούς (με μερικούς περιορισμούς βέβαια σε χαρακτήρες όπως παρενθέσεις κλπ.). Ειδικότερα ο χρήστης μπορεί να κάνει αναζήτηση τύπου “2016-07-29” στο πεδίο date ή “2016” στο πεδίο “year” ώστε το σύστημα να επιστρέψει τραγούδια τα οποία κυκλοφορήσαν εκείνη την περίοδο. Το σύστημα επιπλέον κρατάει ένα ιστορικό στο οποίο φαίνονται προηγούμενες αναζητήσεις ωστόσο το ιστορικό δεν χρησιμοποιείται για να προτείνει το σύστημα στον χρήστη εναλλακτικά ερωτήματα καθώς υπήρχε μεγάλο πρόβλημα με το GUI.

Παρουσίαση αποτελεσμάτων:

Μετά την αναζήτηση τα αποτελέσματα παρουσιάζονται στο ίδιο Interface σε διάταξη ανάλογα με την συνάφεια της ερώτησης ανά 10. Πατώντας την επιλογή “Load More” το σύστημα παρουσιάζει τα επόμενα 10 αποτελέσματα. Για να μπορεί ο χρήστης να δει που ακριβώς βρέθηκε η λέξη ή φράση που εισήγαγε ,το σύστημα εκτός από τον τίτλο του τραγουδιού, το όνομα του τραγουδιστή αλλά και την ημερομηνία κυκλοφορίας εμφανίζει επίσης ένα συγκεκριμένο διάστημα από τους στοίχους του τραγουδιού στο οποίο βρίσκεται ο ορός αναζήτησης και για να είναι πιο εμφανές ο ορός είναι highlighted. Αξίζει να αναφερθεί ότι αν η φράση που εισήγαγε ο χρήστης στο σύστημα δεν παρουσιάζεται σε κανένα από τα .csv αρχεία τότε το σύστημα επιστρέφει ως αποτελέσματα τραγούδια στα οποία εμφανίζεται μια ή περισσότερες από τις λέξεις προς αναζήτηση μεμονωμένες. Ως προς την ομαδοποίηση ο χρήστης έχει μόνο την δυνατότητα να παρουσιάζονται τα αποτελέσματα με βάση την αλφαβητική σειρά του ονόματος των τραγουδιστών και όχι με βάση την συνάφεια ωστόσο εσωτερικά ανά ομάδα τα αποτελέσματα θα βρίσκονται σε διάταξη ανάλογα με την σινάφια του ερωτήματος.

Interface:

Για την αλληλεπίδραση του χρήστη με το σύστημα αλλά και την παρουσίαση αποτελεσμάτων υπάρχει υλοποιημένο ένα γραφικό περιβάλλον (GUI). Πιο ειδικά το Interface αποτελείται από μια βασική οθόνη. Στην οθόνη αυτή υπάρχει η μπάρα αναζήτησης, ένα dropdown στο οποίο ο χρήστης επιλεγεί το πεδίο αναζήτησης, ένα κουμπί οπού προβάλει το ιστορικό αναζήτησης, ένα κουμπί το οποίο ταξινομεί τα αποτελέσματα με βάση το όνομα του τραγουδιστή, ένα κουμπί το οποίο προβάλει τα επόμενα 10 αποτελέσματα και ένα πάνελ στο οποίο εμφανίζονται τα αποτελέσματα αναζήτησης. Για να μπορώ να προβάλω τα αποτελέσματα της αναζήτησης στο UI δημιουργώ ένα νέο αντικείμενο SearchIndex και καλώ την μέθοδο search η οποία διαβάζει τα .csv αρχεία, δημιουργεί το ευρετήριο και εκτελεί την αναζήτηση. Αφού είναι έτοιμα τα αποτελέσματα, αποθηκεύω σε μια μεταβλητή snippet την περιοχή οπού βρέθηκε για πρώτη φορά ο ορός

προς αναζήτηση και όχι όλα τα lyrics ώστε να είναι πιο εύκολο για τον χρήστη να δει που εντοπίστηκε η λέξη-φράση που αναζητεί. Έπειτα για κάθε αποτέλεσμα της αναζήτησης παρουσιάζεται ο τίτλος του τραγουδιού, ο καλλιτέχνης, η ημερομηνία κυκλοφορίας αλλά και το snippet στο οποίο με την χρήση της μεθόδου Highlighter ο όρος αναζήτησης εμφανίζεται τονισμένος.