Ιωάννης Γιαννακός ΑΜ:4970
Απόλλωνας-Πέτρος Καλλιπολίτης ΑΜ:4963

Ανάκτηση πληροφορίας

Αναφορά εργασίας

Ακαδημαϊκό Έτος 2022-2023

Link για το git_hub: https://github.com/Apollon-Peter/Anaktisi.git

Το σύστημα που έχουμε υλοποιήσει, έχει σκοπό να δημιουργεί κατάλληλα μία συλλογή δεδομένων για τραγούδια και ύστερα, να δίνουμε την δυνατότητα στον χρήστη να αναζητά σε αυτή. Η αναζήτηση υποστηρίζει την χρήση φίλτρων και την ομαδοποίηση των αποτελεσμάτων με σκοπό την διευκόλυνση της εύρεσης της επιθυμητής πληροφορίας από τον χρήστη.

Η συλλογή δεδομένων μας, αποτελείται από ένα αρχείο της μορφής csv με το όνομα Eminem.csv το οποίο περιέχει πάνω από 500 τραγούδια του Eminem. Συγκρκιρμένα, συμπεριλαμβάνει τον τίτλο, το όνομα του καλλιτέχνη, το άλμπουμ, την χρονιά αλλά και την ημερομηνία κυκολοφορίας, καθώς και τους στίχους του κάθε τραγουδιού. Εκτός από αυτά τα πεδία, τα οποία αναφέρθηκαν με την σειρά που βρίσκονται και στο αρχείο, υπάρχει και ένα ακόμα πεδίο πριν τις πληροφορίες κάθε τραγουδιού, το οποίο αναπαριστά το νούμερο θέσης του τραγουδιού μέσα στο αρχείο. Αυτό το πεδίο όμως δεν το αξιοποιούμε, ούτε το επεξεργασόμαστε, με κάποιον τρόπο. Για την εύρεση αυτού του αρχείου, αναζητήσαμε για συλλογές δεδομένων στο Kaggle.com, μία ιστοσελίδα που παρέχει τους χρήστες της με συλλογές δεδομένων, η οποία έχει σκοπό να βοηθήσει στην επιστήμη των δεδομένων (data science).

Κατά την επεξεργασία του αρχείου αυτού, με σκοπό να δημιουργήσουμε μία κατάλληλη συλλογή δεδομένων, εμείς επεξεργαζόμαστε από μία γραμμή τη φορά. Συγκεκριμένα, επειδή όλες οι πληροφορίες του κάθε τραγουδιού βρίσκονται σε μία γραμμή, εμείς δημιουργούμε αντικείμενα τύπου document, για κάθε τραγούδι, στα οποία προσθέτουμε πεδία. Τα πεδία αυτά συγκρατούν τις πληροφορίες κάθε τραγουδιού που αναφέρθηκαν προηγουμένως και δημιουργούμε από ένα , ή δύο πεδία για κάθε είδος πληροφορίας. Δύο, επιλέγουμε να δημιουργήσουμε στα πεδία όπου εφαρμόζουμε stemming, με σκοπό να χρησιμοποιήσουμε το ένα για την αναζήτηση και το άλλο για την εκτύπωση των πληροφοριών. Στα υπόλοιπα πεδία που δεν εφαρμόζουμε stemming ή που δεν τα χρησιμοποιούμε για την εκτύπωση των πληροφοριών, δημιουργούμε μόνο από ένα πεδίο για κάθε είδος πληροφορίας. Με αυτόν τον τρόπο επεξεργασίας του αρχείου, έχουμε τη δυνατότητα αργότερα να εκτελούμε αναζήτηση μόνο σε συγκεκριμένα πεδία, δηλαδή μόνο σε συγκεκριμένες κατηγορίες πληροφοριών. Όλες τις πληροφορίες τις αποθηκεύουμε σε πεδία κειμένου με εξαίρεση την χρονιά κυκλοφορίας του κάθε τραγουδιού. Εκεί χρησιμοποιούμε πεδία τύπου NumericDocValues και Stored με σκοπό να τα χρησιμοποιήσουμε για ομαδοποίηση των αποτελεσμάτων με βάση την χρονιά κυκλοφορίας τους, σε περίπτωση που το επιθυμεί αυτό ο χρήστης. Για την υλοποίηση των παραπάνω, χρησιμοποιούμε την lucene και τα κατάλληλα εργαλεία που μας παρέχει, όπως το ίδιο κάνουμε και για την δημιουργία του ευρετηρίου μας. Πιο αναλυτικά, χρησιμοποιούμε τον Standard Analyzer και θέτουμε ως stop words τις προεπιλεγμένες για την αγγλική γλώσσα, με την βοήθεια της lucene. Αυτή η διαδικασία θα ακολουθηθεί για κάθε αρχείο

που βρίσκεται μέσα στον φάκελο με τα input αρχεία, αλλά στην περίπτωσή μας θα εκτελεστεί μόνο μία φορά.

Για να αναζητήσει ο χρήστης κάποιο τραγούδι, του δίνεται η επιλογή να ψάξει οποιαδήποτε πληροφορία ενός τραγουδιού, όπως το όνομα του καλλιτέχνη, τον τίτλο, το άλμπουμ, την χρονιά αλλά και την ημερομηνία κυκλοφορίας, ακόμα και τους στίχους του. Του παρέχεται η δυνατότητα μέσα από ένα drop-down menu να εφαρμόσει κατάλληλα φίλτρα που να περιορίζουν την αναζήτησή του μόνο σε συγκεκριμένα πεδία, δηλαδή σε μία συγκεκριμένη κατηγορία πληροφοριών από αυτές που μόλις αναφέραμε. Αφού ο χρήστης πραγματοποιήσει κάποια αναζήτηση, ο όρος για τον οποίο εκτέλεσε την αναζήτηση, αποθηκεύεται στο ιστορικό αναζητήσεων μαζί με το φίλτρο που είχε επιλεχθεί. Κάθε όρος αναζήτησης και φίλτρο που χρησιμοποιήθηκε για την κάθε μία, θα είναι διαθέσιμα σε αυτό το παράθυρο μέχρι να τερματιστεί η εφαρμογή. Για την υλοποίηση της αναζήτησης των δεδομένων, χρησιμοποιήσαμε πάλι την lucene δημιουργώντας αναζητητές (searcher) και αναγνώστες (reader) με σκοπό να αναζητούμε με αυτούς τις επιθυμητές πληροφορίες στα ευρετήρια. Επίσης δημιουργούμε αντικείμενα τύπου query με σκοπό να πραγματοποιήσουμε την αναζήτηση σε συγκεκριμένα πεδία μόνο, αλλά και αντικείμενα τύπου sort για να ομαδοποιήσουμε τα αποτελέσματα. Με την κλήση της συνάρτησης searcher δίνουμε σαν ορίσματα το query και το sort κάθε φορά όπως και τον αριθμό αποτελεσμάτων που θέλουμε να δεχτούμε.

Όσο αφορά την παρουσίαση των αποτελεσμάτων, το σύστημά μας εκτυπώνει τα αποτελέσματα που συνάδουν με την αναζήτηση του χρήστη, ανά δέκα την φορά, δίνοντάς του την επιλογή να προχωρήσει στα επόμενα δέκα με το πάτημα του αντίστοιχου κουμπιού. Κάθε φορά που επιλέγουμε το πλήκτρο “Next 10 results”, με σκοπό να πάρουμε τα επόμενα δέκα αποτελέσματα , αυξάνουμε κατά δέκα δύο μεταβλητές τις οποίες χρησιμοποιούμε για να τυπώσουμε μόνο τα επιθυμητά δέκα αποτελέσματα. Πάνω από τα αποτελέσματα τυπώνεται ένα μήνυμα που ενημερώνει τον χρήστη με ποιο φίλτρο πραγματοποίησε την αναζήτησή του. Επιπλέον στα αποτελέσματα είναι τονισμένοι οι χαρακτήρες που ταίριαξαν με την αναζήτηση, για διευκόλυνση του χρήστη, και τυπώνουμε κάθε φορά τον τίτλο, τον καλλιτέχνη, το άλμπουμ και την χρονιά κυκλοφορίας, χωρισμένα με αυτό το σύμβολο “|” αλλά και τους στίχους του τραγουδιού (στην επόμενη γραμμή), το οποίο βρέθηκε ως hit. Επίσης δίνεται η επιλογή στον χρήστη μέσω ενός διακόπτη, να ενεργοποιήσει την ομαδοποίηση των αποτελεσμάτων με βάση την χρονολογία έκδοσής τους σε αύξουσα σειρά. Σε περίπτωση που δεν υπήρχαν αποτελέσματα για την αναζήτηση που πραγματοποιήθηκε, τυπώνεται το μήνυμα “No results were found!” ενώ αφού τελειώσουν τα αποτελέσματα μίας αναζήτησης τυπώνεται το μήνυμα “ No more results were found!”.

Για την υλοποίηση του interface του συστήματος, χρησιμοποιήσαμε το έτοιμο γραφικό περιβάλλον της java (GUI). Σε αυτό προσθέσαμε στο πάνω μέρος του, μία μπάρα αναζήτησης, ένα κουμπί “Search”, ένα drop-down menu από το οποίο επιλέγεις τα φίλτρα της αναζήτησης, το κουμπί “Next 10 results” με το οποίο προχωράς στα επόμενα 10 αποτελέσματα, το κουμπί “Search History” το οποίο σου ανοίγει ένα παράθυρο στο οποίο αναγράφονται οι όροι αναζήτησης και τα φίλτρα που χρησιμοποίησε ο χρήστης σε κάθε αναζήτησή του, και έναν διακόπτη “Sort by Year” ο οποίος αφού ενεργοποιηθεί, ταξινομεί τα αποτελέσματα με κριτήριο την χρονολογία κυκλοφορίας τους σε αύξουσα σειρά. Τέλος προσθέσαμε την περιοχή εμφάνισης των αποτελεσμάτων, την οποία αργότερα χρησιμοποιούμε για την εκτύπωση όλων των αποτελεσμάτων που αναφέραμε προηγουμένως.

Ως βιβλιοθήκες για το σύστημά μας, χρησιμοποιήσαμε τις ακόλουθες της lucene:

lucene-analysis-common-9.5.0.jar

lucene-core-9.5.0.jar

lucene-queryparser-9.5.0.jar

,αλλά και την commons-io-2.11.0 για να μπορέσουμε να διαγράψουμε τα περιεχόμενα του φακέλου με τα ευρετήρια, κάθε φορά που γίνεται εκκίνηση του συστήματός μας.