Information Retrieval 2022

Βουρδουγιάννης Δημήτριος 4326

Νικόλαος Παραλίας Ψιλοδημητρακόπουλος 4202

❖ Λειτουργικό σύστημα και περιβάλλον

Η συγκεκριμένη εργασία υλοποιήθηκε στο λειτουργικό περιβάλλον Windows 10 (64 bit) και στην πλατφόρμα Intellij Idea της εταιρίας Jetbrains. Το project δοκιμάστηκε και στην πλατφόρμα του eclipse και είναι λειτουργικό.

❖ Ορισμός – Τι σημαίνει Apache Lucene;

Η Apache Lucene είναι μια μηχανή αναζήτησης υψηλής απόδοσης με την έννοια του "εγγράφου που περιέχει πεδία κειμένου" στην κεντρική λογική αρχιτεκτονική του. Αυτό προσφέρει μεγάλη ευελιξία και επιτρέπει στο Lucene API να γίνει ανεξάρτητο από οποιαδήποτε μορφή αρχείου. Οποιοδήποτε κείμενο από μορφές όπως MS Word, HTML, XML, PDF και OpenDocument (ή αρχείο csv όπως στην περίπτωση μας) μπορεί να ευρετηριωθεί όσο μπορούν να εξαχθούν οι πληροφορίες κειμένου, πράγμα που σημαίνει ότι δεν μπορεί να κάνει τίποτα με εικόνες.

❖ Προ επεξεργασία δεδομένων - Ευρετηριοποίηση

Εμείς χρησιμοποιήσαμε τη Lucene για να υλοποιήσουμε ένα σύστημα αναζήτησης άρθρων σχετικά με ταινίες. Αρχικά, επισκεφτήκαμε το σύνδεσμο "https://www.kaggle.com/datasets/sankha1998/tmdb-top-10000-popular-movies-dataset?select=TMDb_updated.CSV" και κατεβάσαμε μία από τις έτοιμες συλλογές που μας προτείνατε. Το αρχείο που κατεβάσαμε είναι τύπου csv, ανοίγει με κωδικοποίηση UTF-8 και τα περιεχόμενά του είναι ταινίες από όλο τον κόσμο. Πιο συγκεκριμένα, περιέχονται 10.000 ταινίες από τις οποίες κρατήσαμε τις πρώτες 2016 (χωρίς συγκεκριμένο κριτήριο επιλογής) και 6 πεδία από τα οποία κρατήσαμε τα 5 που αποφασίσαμε ότι θα μας βοηθούσαν στην αναζήτηση ταινιών, τα πεδία αυτά είναι: title (ο τίτλος της ταινίας), overview (σύντομη περιγραφή της ταινίας), original_language (η γλώσσα της ταινίας), vote_count (το σύνολο των ατόμων που την έχει βαθμολογήσει) και vote_average (ο μέσος όρος των βαθμολογιών της).

Για την κατασκευή του ευρετηρίου χρησιμοποιήσαμε τον StandardAnalyzer, με σκοπό να γίνεται απαλοιφή των stop words. Στο συγκεκριμένο κομμάτι θέλαμε να επεκταθούμε εφαρμόζοντας κάποιες λειτουργίες που μας προτείνετε όπως η διόρθωση τυπογραφικών λαθών ή η επέκταση ακρωνύμων, κάτι το οποίο εν τέλει δεν έγινε λόγω έλλειψης χρόνου.

❖ Αναζήτηση

Η υλοποίηση μας υποστηρίζει δύο είδη αναζήτησης, την απλή και τη σύνθετη αναζήτηση.

- Απλή αναζήτηση:

Όταν ο χρήστης θέλει να ψάξει μία ταινία ακόμα κι αν δεν έχει αρκετές πληροφορίες για αυτήν στρέφεται προς την απλή αναζήτηση. Στην μπάρα αναζήτησης γράφει τη λέξη κλειδί που περιέχεται στον τίτλο της ταινίας ή στην σύντομη περιγραφή της ταινίας και αν η συγκεκριμένη λέξη υπάρχει σε μία ή περισσότερες ταινίες θα εμφανιστούν τα ανάλογα αποτελέσματα.

- Σύνθετη αναζήτηση:

Στην περίπτωση που ο χρήστης θέλει να ψάξει ταινίες με συγκεκριμένα χαρακτηριστικά όπως για παράδειγμα η γλώσσα της ταινίας ή ο μέσος όρος των βαθμολογιών της θα στραφεί στη σύνθετη αναζήτηση και θα συμπληρώσει τα κατάλληλα πεδία που θέλει. Δηλαδή, θα συμπληρώσει το “Original Language” γράφοντας “en” αν θα ήθελε να εμφανιστούν αγγλικές ταινίες ή το “Votes Average” γράφοντας «10» αν ήθελε να εμφανιστούν μόνο οι ταινίες που έχουν μέσο όρο βαθμολογίας 10. Στην περίπτωση που συμπληρώσει και τα δύο πεδία θα τον έβγαζε το συνδυασμό αυτών, δηλαδή μόνο αγγλικές ταινίες με βαθμολογία 10.

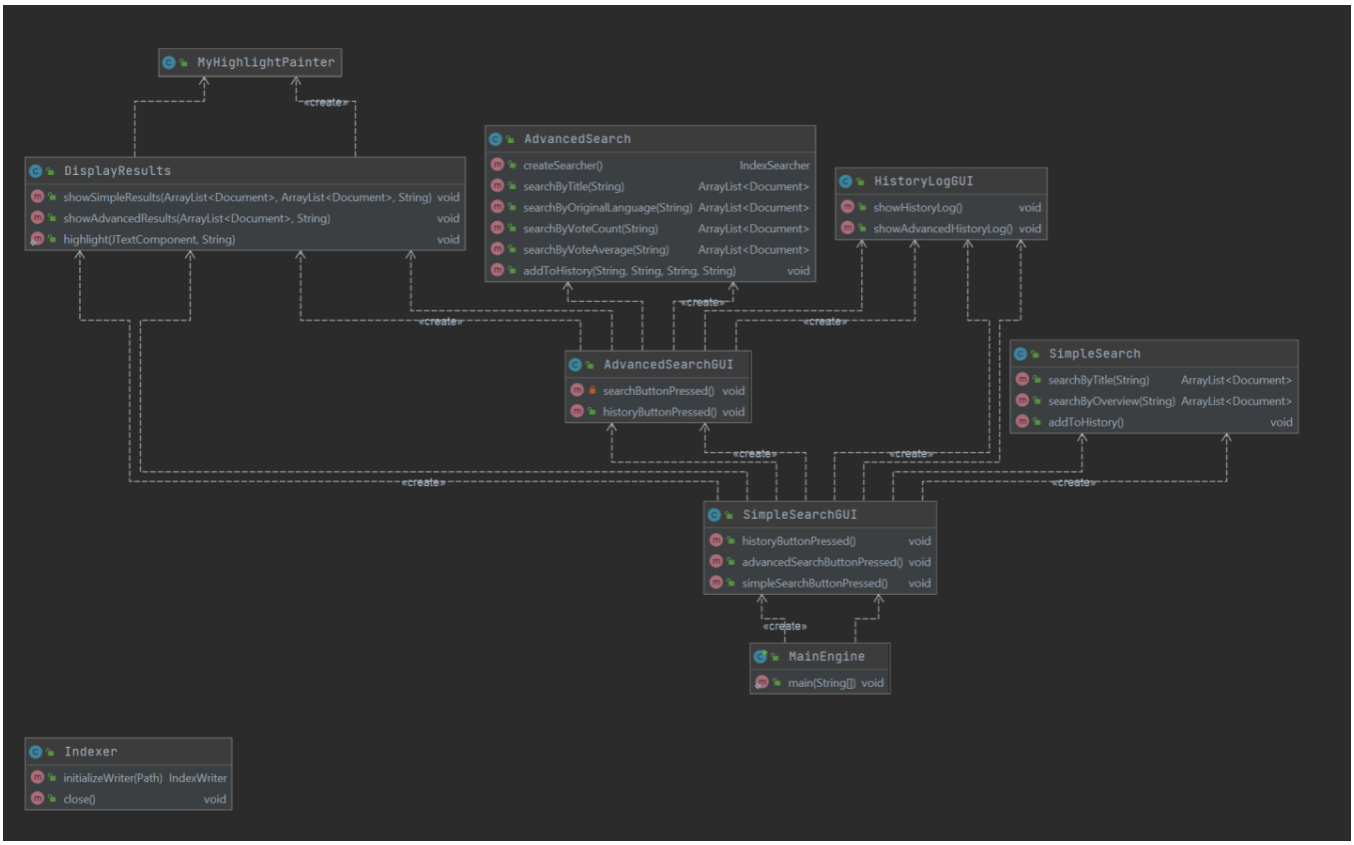
- Ιστορικό Αναζήτησης:

Σε οποιαδήποτε φάση της αναζήτησης ο χρήστης μπορεί να πατήσει το κουμπί “History” και να δει τις προηγούμενες αναζητήσεις του, είτε για την απλή, είτε για τη σύνθετη.

❖ Παρουσίαση αποτελεσμάτων

Αφού ο χρήστης πατήσει το κουμπί της αναζήτησης, είτε είναι απλή είτε σύνθετη, παρουσιάζονται τα αποτελέσματα με όλα τα χαρακτηριστικά της ταινίας, δηλαδή όλα τα πεδία (title, overview, original_language, vote_count, vote_average) με την αντίστοιχη πληροφορία. Πιο συγκεκριμένα εμφανίζονται οι ταινίες σε μορφή παπύρου με τη δυνατότητα ο χρήστης να κυλάει τον δρομέα (κάνει σκρολ) μεταξύ των αποτελεσμάτων. Επιπλέον, οι λέξεις κλειδιά που έχει αναζητήσει ο χρήστης είναι τονισμένες με κίτρινο χρώμα μόνο στα πεδία του τίτλου και της σύντομης περιγραφής, καθώς έτσι θεωρήσαμε ότι θα είναι πιο ορθό

❖ UML διάγραμμα



❖ Σχεδιασμός κώδικα

Στην υλοποίησή μας έχουμε δημιουργήσει 5 πακέτα.

- **engine:** Περιέχει την κλάση MainEngine όπου περιέχει τη μέθοδο main.
- **index:** Περιέχει την κλάση Indexer στην οποία γίνεται το indexing του αρχείου και δημιουργείται το ευρετήριο με τη βοήθεια της Lucene.
- **modelResults:** Περιέχει δύο κλάσεις, την κλάση MyHighlightPainter, η οποία είναι βοηθητική κλάση της DisplayResults, και τη κλάση DisplayResults, όπου με αυτές εμφανίζονται τα αποτελέσματα των ταινιών στον χρήστη.
- **searchengines:** Περιέχει δύο κλάσεις, την SimpleSearch και την AdvancedSearch, στις οποίες υλοποιούνται τα δύο συστήματα αναζήτησης.
- **view:** Περιέχει τρεις κλάσεις, την SimpleSearchGUI, την AdvancedSearchGUI και την HistoryLogGUI, όπου υλοποιείται το γραφικό περιβάλλον της απλής αναζήτησης, της σύνθετης αναζήτησης και του ιστορικού.

❖ Παρατηρήσεις

1. Στη σύνθετη αναζήτηση θέλαμε ο χρήστης να επιλέγει πεδίο για να ψάχνει vote average και vote count, παραδείγματος χάρη vote average από 5,5 και πάνω. Δεν μπορέσαμε να υλοποιήσουμε την ανισότητα, ωστόσο μπορεί να γίνεται κανονικά αναζήτηση με τιμή, δηλαδή για την τιμή που θα εισάγει ο χρήστης θα επιστραφούν ταινίες μόνο με αυτή την τιμή (έτσι κι αλλιώς υπάρχει πάντα η δυνατότητα να μην βάλει τίποτα σε αυτά τα πεδία ο χρήστης).
2. Όπως προαναφέρθηκε το πρόγραμμά μας κρατάει ιστορικό αναζητήσεων, αλλά υστερεί στο να δίνει περισσότερες επιλογές στον χρήστη για το τι θέλει να ψάξει με βάση το ιστορικό. Πιο συγκεκριμένα προσπαθήσαμε να υλοποιήσουμε έναν τρόπο αναζήτησης, ο οποίος να προβλέπει τις λέξεις κλειδιά που εισάγει ο χρήστης αλλά και να του προτείνει λέξεις κλειδιά με βάση το ιστορικό αναζητήσεων. Ωστόσο, δεν το καταφέραμε διότι είχαμε θέματα ασυμβατότητας με τις βιβλιοθήκες της java swing.
3. Κατά την παρουσίαση των αποτελεσμάτων έπρεπε να υπάρχει μία τουλάχιστον επιλογή ταξινόμησης. Προσπαθήσαμε να χρησιμοποιήσουμε τη βιβλιοθήκη Sort της Lucene, αλλά η χρήση της αποδείχτηκε περίπλοκη επειδή δεν βρήκαμε πολλά και χρήσιμα παραδείγματα εφαρμογής της.