Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής
Πολυτεχνική Σχολή - Πανεπιστήμιο Ιωαννίνων

Εργασία : Μηχανή αναζήτησης ταινιών 'MovieIt'

Φάση 2 : Τελική αναφορά της εργασίας

Αθανάσιος Τόμπρας ΑΜ : 3345
Βασίλειος Βογιάννου ΑΜ : 3193

Περιεχόμενα

Συλλογή εγγράφων

Για τη συλλογή εγγράφων χρησιμοποιήθηκε το προτεινόμενο dataset του Kaggle με τις 10000 πιο δημοφιλείς ταινίες από τις οποίες χρησιμοποιήθηκαν οι πρώτες 5000. Παρατίθεται ο σύνδεσμος: https://www.kaggle.com/datasets/ sankha1998/tmdb-top-10000-popular-movies-dataset
Τα πεδία που λήφθηκαν υπ΄όψιν είναι τέσσερα από τα συνολικά έξι πεδία που περιέχει το dataset για κάθε ταινία ενώ το format των δεδομένων μετατράπηκε από .csv σε .json καθώς θεωρήθηκε πιο εύκολη η επεξεργασία των κειμένων στο συγκεκριμένο format. Ως πεδία του dataset ορίζονται τα «title, overview, original language, vote average».

Πιο αναλυτικά:

- Title: Αποτελεί τον τίτλο της κάθε ταινίας (String - Text)
- Overview: Μία μικρή περίληψη της κάθε ταινίας (String - Text)
- Original language: Η γλώσσα στην οποία γυρίστηκε η ταινία (String - Text)
- Vote average: Ο μέσος όρος των ψήφων για την κάθε ταινία (Float - Numeric)

Τέλος, παρατηρήθηκαν διπλότυποι τίτλοι και missing values στο πεδίο overview του dataset με αποτέλεσμα την εξαίρεση των συγκεκριμένων γραμμών από τα δεδομένα μας για μεγαλύτερη ευκολία στο χειρισμό τους.

Μετατροπή format

Για τη μετατροπή του format του .csv αρχείου σε .json δημιουργήθηκε το παρακάτω python script σε jupyter notebook.

```
import csv
import json

def csv_to_json(csvFilePath, jsonFilePath):
    jsonArray = []

    #read csv file
    with open(csvFilePath, encoding='utf-8') as csvf:
        #load csv file data using csv library's dictionary reader
        csvReader = csv.DictReader(csvf)

        #convert each csv row into python dict
        for row in csvReader:
            #add this python dict to json array
            jsonArray.append(row)

    #convert python jsonArray to JSON String and write to file
    with open(jsonFilePath, 'w', encoding='utf-8') as jsonf:
        jsonString = json.dumps(jsonArray, indent=4)
        jsonf.write(jsonString)

csvFilePath = r'C:\Users\Thanos\Desktop\12TH SEMESTER\ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ\PROJECT\csvfile.csv'
jsonFilePath = r'C:\Users\Thanos\Desktop\12TH SEMESTER\ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ\PROJECT\jsonfile.json'
csv_to_json(csvFilePath, jsonFilePath)
```

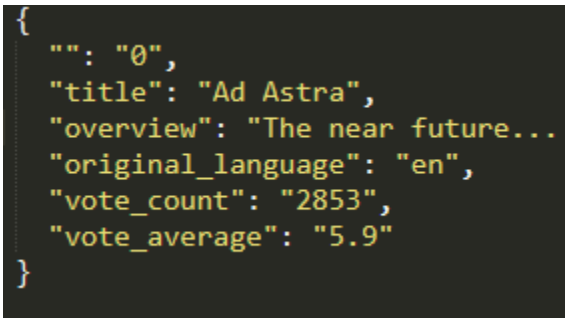
Στη συνέχεια χωρίσαμε το JSON αρχείο σε ξεχωριστά JSON documents με τα πεδία τους ώστε να είναι πιο εύκολη η ανάγνωση τους από τον Parser του συστήματος και η επεξεργασία τους. Υλοποιήθηκε αντίστοιχο python script για τη δημιουργία των json documents.

```
import json

docs = json.load(open('C:\\Users\\Thanos\\Desktop\\12TH SEMESTER\\ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ\\PROJECT\\jsonfile.json'))

for ii, doc in enumerate(docs):
    with open('doc{}.json'.format(ii), 'w') as out:
        json.dump(doc, out, indent=2)
```

Μετά την τελική μετατροπή το κάθε json document έχει την παρακάτω μορφή με τα πεδία του και τις αντίστοιχες τιμές τους και αποθηκεύονται στο τοπικό directory στον φάκελο ‘JSONS’ ώστε να τα διαβάσει στη συνέχεια το σύστημα μας.



Στόχος και λειτουργικότητα του συστήματος

Ο στόχος του συστήματος είναι η υλοποίηση μιας αποδοτικής μηχανής αναζήτησης αξιοποιώντας την βιβλιοθήκη Apache Lucene. Το σύστημα θα επεξεργάζεται τα JSONS θα δημιουργεί documents και θα δημιουργεί μια βιβλιοθήκη αρχείων αναγνώσιμων από το βασικό πρόγραμμα, Index μέσω αυτών των απλών αρχείων, και τέλος θα βρίσκει την κατάλληλη ταινία βάσει της λέξης/φράσης αναζήτησης του χρήστη. Τέλος, έχει υλοποιηθεί ένα φιλικό προς τον χρήστη σύστημα γραφικών GUI με σκοπό την εύκολη περιήγηση του στην εφαρμογή.

Δημιουργία των αρχείων

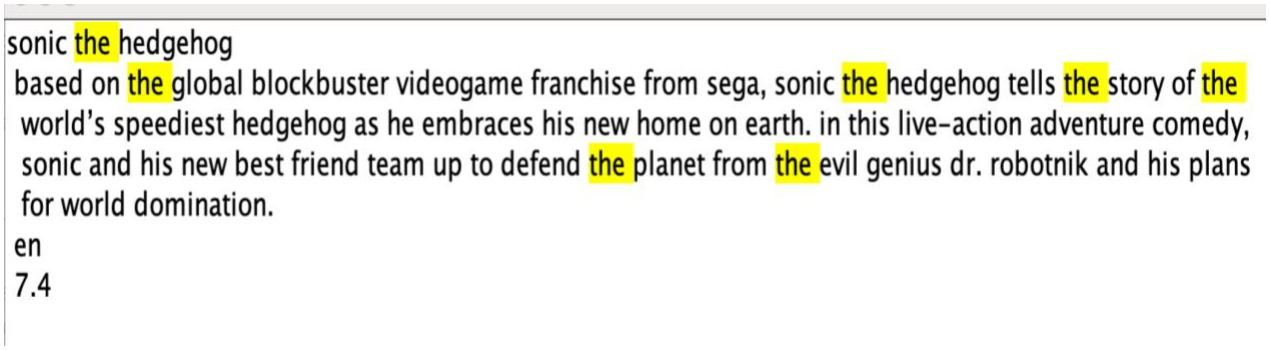
Τα έγγραφα τα οποία έχουν επιλεγεί είναι συγκεντρωμένα στο φάκελο ‘JSONS’ του τοπικού directory. Για την επεξεργασία τους υλοποιήθηκε η κλάση «JSONExtractor». Ο σκοπός της είναι να μετατρέψει τα json documents σε μια πιο απλή μορφή .txt εύκολα αναγνώσιμη από το σύστημα μας για τη δημιουργία των Documents στη συνέχεια. Στην υλοποίηση μας αποθηκεύονται και τα τέσσερα πεδία με τον τίτλο αρχικά, στη συνέχεια την περιγραφή της ταινίας, μετά τη γλώσσα παραγωγής της και τέλος το μέσο όρο ψήφων που απέσπασε από το κοινό. Το κάθε αρχείο .txt φέρει ως όνομα το id της αντίστοιχης ταινίας. Παρακάτω φαίνεται η υλοποίηση του αρχείου με τα 4 πεδία, ένα σε κάθε σειρά.

```
Ad Astra
The near future, a time when both hope and hardships drive humanity to look to the stars and beyond.
en
5.9
```

Έχοντας τα αρχείο μας συρρικνωμένα για εύκολη επεξεργασία, περνάμε στο Documentation phase της Lucene. Εκεί δημιουργούμε πολλαπλά Documents με πεδία “title”,“overview”, “original_language” και “vote_average” που θα αξιοποιούνται στην αναζήτηση. Η δημιουργία αυτών των Documents γίνεται μέσω του Standard Analyzer της Lucene,όπου δημιουργεί τα Index με τη χρήση ενός Index Writer και τα τοποθετεί στο τοπικό Directory “Indexes”, ενώ η συλλογή των Documents γίνεται από τον φάκελο “Documents”.

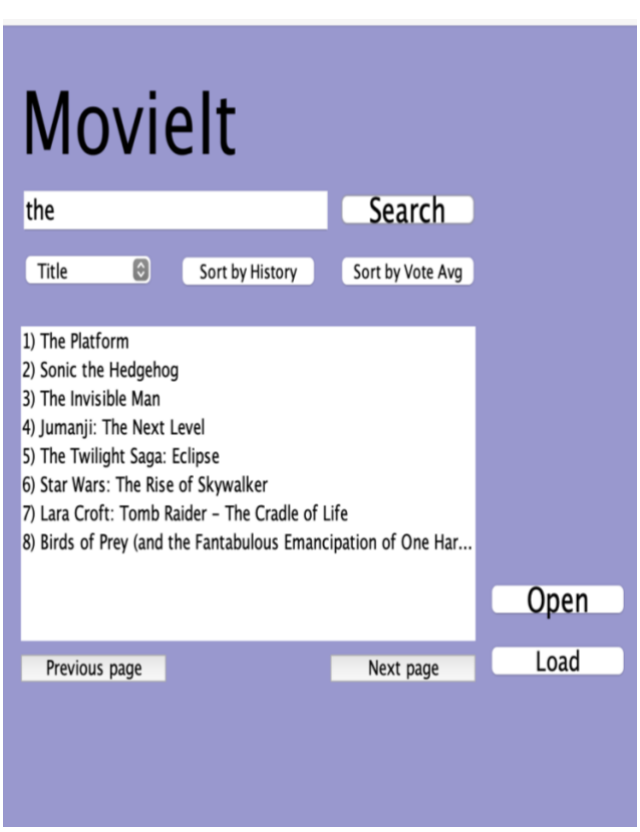
Αναζήτηση

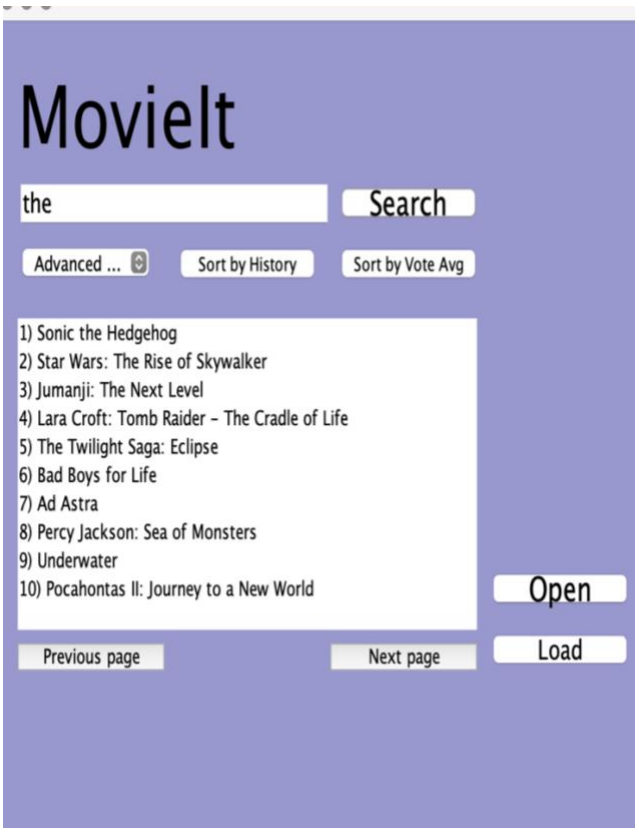
Η διαδικασία αναζήτησης γίνεται μέσω ενός Index Searcher. Αξιοποιώντας τα προηγούμενα Indexes στον φάκελο “Indexes” , καθώς και ένα Query Parser, ελέγχει βάσει του Standard Analyzer τις ταινίες στα Document μας, επιστρέφοντας τις δέκα πρώτες ταινίες(TopDocs) βάσει ενός score που έχουμε καθορίσει. Έχοντας τα κλειδιά των κορυφαίων Documents, δίνεται η δυνατότητα να εντοπιστούν οι ταινίες και να τυπωθούν στον χρήστη πληροφορίες για τα υπόλοιπα πεδία τους πιέζοντας το πλήκτρο «Open» όπως φαίνεται παρακάτω. Υλοποιήθηκε επίσης η λειτουργία υπογράμμισης(highlighting) των λέξεων-κλειδιών που αναζητά ο χρήστης, στο αρχείο που εμφανίζει τις πληροφορίες της ταινίας στην οθόνη.



Ο χρήστης έχει τη δυνατότητα να αναζητήσει τον τίτλο της ταινίας ενώ υπάρχει και η δυνατότητα αναζήτησης λέξεων/φράσεων στα υπόλοιπα πεδία ξεχωριστά. Επιπλέον με την επιλογή «Advanced Search» γίνεται ταυτόχρονη αναζήτηση της λέξης/φράσης σε 3 πεδία (Τίτλος, Περίληψη, Γλώσσα) με βάση την εμφάνιση της λέξης αυτής συνολικά στα πεδία.

Όπως διαπιστώνεται παρακάτω μία αναζήτηση της λέξης «the» στον τίτλο μας δίνει τις ταινίες που περιέχουν «the» στον τίτλο τους τοποθετώντας στην κορυφή αυτές που έχουν λιγότερες λέξεις και πιο νωρίς τη λέξη κλειδί και στη συνέχεια τις επόμενες στην κατάταξη. Ενώ επιλέγοντας το πεδίο «Advanced Search» και πληκτρολογώντας την ίδια λέξη παρατηρούμε ότι η σειρά έχει αλλάξει αφού προσμετρώνται πλέον και οι φορές που βρίσκεται η λέξη κλειδί στην περίληψη ή/και στη γλώσσα.

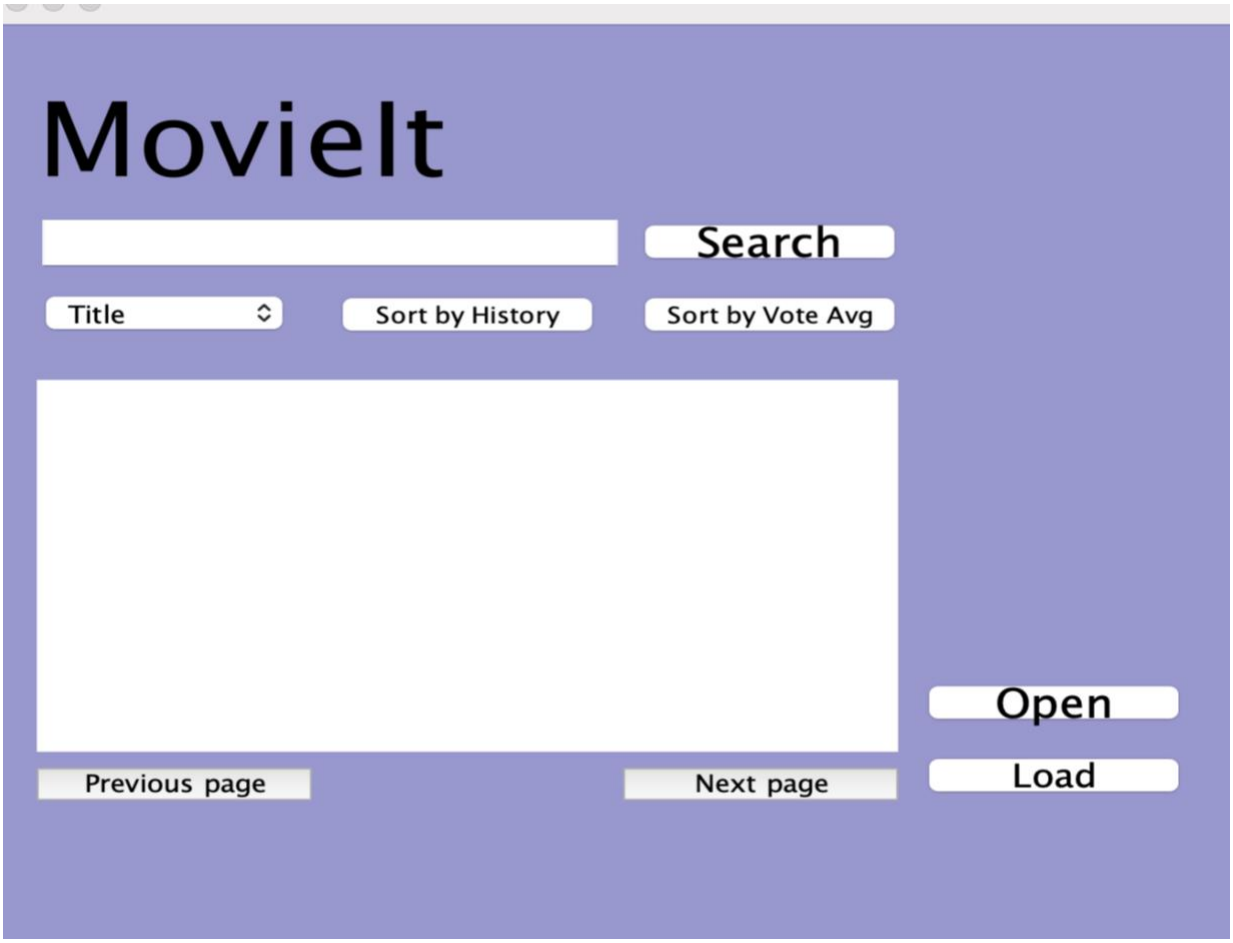




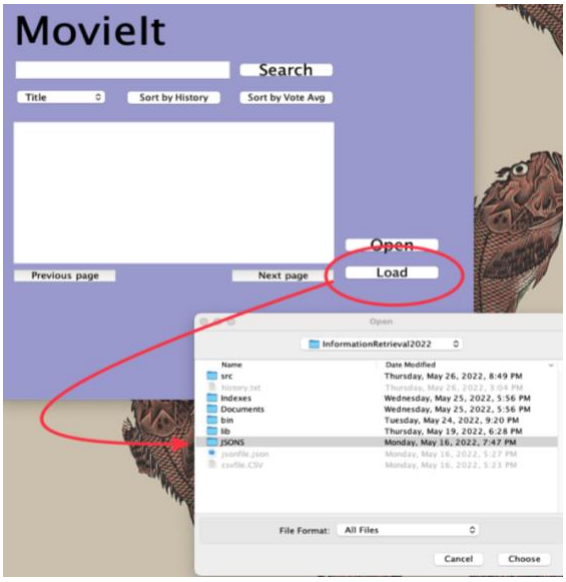
Τέλος, παρέχονται οι επιλογές εμφάνισης των ταινιών βάσει του ιστορικού αναζήτησης λέξεων/φράσεων σε όλα τα πεδία καθώς και βάσει του μέσου όρου βαθμολογίας που έχουν αποσπάσει από το κοινό. Όσον αφορά την ταξινόμηση βάσει ιστορικού, με την πρώτη αναζήτηση κάποιας λέξης/φράσης δημιουργείται ένα αρχείο με όνομα «history.txt» στο τοπικό directory στο οποίο καταγράφονται όλες οι αναζητήσεις των λέξεων-κλειδιών. Στη συνέχεια, πατώντας το κουμπί «Sort By History» και επιλέγοντας την αντίστοιχη κατηγορία αναζήτησης εμφανίζονται οι τίτλοι της ταινίας που είναι πιο σχετικοί με την πιο δημοφιλή αναζήτηση του χρήστη βάσει ιστορικού στο πεδίο αυτό. Για να γίνει ταξινόμηση σύμφωνα με τη βαθμολογία κάθε ταινίας αρκεί ο χρήστης να επιλέξει το πεδίο αναζήτησης και αφού του εμφανιστούν οι αντίστοιχοι τίτλοι πατώντας «Sort By Vote Avg» ταξινομούνται σε φθίνουσα σειρά οι τίτλοι της ταινίας.

Τελικό GUI και παρουσίαση αποτελεσμάτων

Η μηχανή αναζήτησης παρέχει στον χρήστη ένα στοιχειώδες παράθυρο GUI με μερικάκουμπιά για τις διάφορες λειτουργίες του συστήματος όπως φαίνεται παρακάτω.



Αρχικά ο χρήστης επιλέγει τη φόρτωση των json αρχείων επιλέγοντας την εντολή «Load» και δίνοντας το μονοπάτι που βρίσκεται ο φάκελος των αρχείων, το σύστημα δημιουργεί τα indexes και τα documents στους αντίστοιχους φακέλους. Δίνεται η δυνατότητα να γίνουν overwrite τα αρχεία στους φακέλους σε περίπτωση που υπάρχουν ήδη προηγούμενα. Το σύστημα εμφανίζει τις ανάλογες ερωτήσεις.



Έπειτα, ανάλογα με το ερώτημα που τίθεται, η εφαρμογή παρουσιάζει τα αποτελέσματα σε διάταξη με βάση τη συνάφεια τους με το ερώτημα. Η προεπιλεγμένη διάταξη όσον αφορά τα πεδία αναζήτησης θα εμφανίζει τους δέκα πιο σχετικούς τίτλους ταινιών σύμφωνα με την αντίστοιχη λέξη/φράση κλειδί και θα υπάρχει δυνατότητα προβολής πληροφοριών των υπολοίπων πεδίων της κάθε ταινίας. Παρέχεται η δυνατότητα μετάβασης στην επόμενη δεκάδα ταινιών καθώς και στην προηγούμενη χρησιμοποιώντας τα πλήκτρα «Next page» και «Previous page».

Τεχνικές πληροφορίες

Η υλοποίηση του προγράμματος έγινε με τη χρήση της πλατφόρμας Eclipse σε γλώσσα προγραμματισμού Java(Version 18.0) και περιβάλλοντα Windows και MacOS.
Στη συγκεκριμένη υλοποίηση το πρόγραμμα τρέχει σε λογισμικό Windows. Για να υποστηρίζεται από λογισμικό MacOS και Linux πρέπει να γίνουν οι απαραίτητες τροποποιήσεις στον ορισμό των Paths όπως ορίζονται παρακάτω:

```
public void writeIndexes(String path)
{
    if (iwriter == null) { return; }
    try
    {
        //Source file of our document library
        String source = Paths.get("").toAbsolutePath().toString() + "/" + path;
        File dir = new File(source);
```

```
public void writeIndexes(String path)
{
    if (iwriter == null) { return; }
    try
    {
        //Source file of our document library
        String source = Paths.get("").toAbsolutePath().toString() + "\\" + path;
        File dir = new File(source);
```

Οι τροποποιήσεις πρέπει να εφαρμοστούν στα εξής αρχεία :Indexer.java , JSONExtractor.java, Searcher.java και UserInterface.java