Ανάκτηση πληροφορίας

Μέλη ομάδας:

- Βλαχόπουλος Λάμπρος , 2948
- Τζάρας Αναστάσιος , 3343

Στα πλαίσια της εργασίας για το μάθημα ανάκτηση πληροφορίας θα ασχοληθούμε με την έτοιμη συλλογή δεδομένων που βρήκαμε στις διαφάνειες και αφορά τις ταινίες και σειρές του netflix.

Η συλλογή βρίσκεται για αρχή σε ένα .csv αρχείο και αποτελείται από 12 χαρακτηριστικά για κάθε ταινία-σειρά τα οποία παρατηρήσαμε ότι διαχωρίζονται με κόματα (,). Τα χαρακτηριστικά που περιλαμβάνονται στην συλλογή αφορούν :

- show_id
- Type
- Title
- Director
- Cast
- Country
- Date_added
- Release_year
- Rating
- Duration
- Listen_in
- Description

Αυτό που σκεφτήκαμε για αρχή είναι να κάνουμε parsing κάθε γραμμή και ταυτόχρονα για κάθε γραμμή tokenizing για να έχουμε τις λεκτικές μονάδες και να αναγνωρίσουμε για κάθε εγγραφή τα επιμέρους χαρακτηριστικά της. Για παράδειγμα σκεφτόμαστε να κάνουμε μια κλάση Document με κάποια πεδία από τα 12 χαρακτηριστικά ή και όλα όσα έχει η συλλογή

και αφού διαβάσουμε όλο το αρχείο .csv θα έχουμε δημιουργήσει όλα τα document με πεδία αρχικοποιημένα όπως τα βλέπουμε στην συλλογή.

Παρατηρήσαμε ότι για κάποιες εγγραφές για ταινίες ή σειρές δεν υπάρχουν όλα τα χαρακτηριστικά δηλαδή για κάποια ταινία μπορεί να μην υπάρχει η πληροφορία για το cast. Θα προσπαθήσουμε να μην δημιουργεί πρόβλημα στην δομή ενημερώνοντας τα κατάλληλα πεδία με τις αντίστοιχες τιμές (‘null’) . Επίσης θα είμαστε ιδιαίτερα προσεκτικοί στην συλλογή των λεκτικών μονάδων καθώς παρατηρήσαμε κυρίως στο description υπάρχουν (,) που δεν διαχωρίζουν τις μονάδες σαν χαρακτηριστικά αλλά γίνεται χρήση του για γλωσσικούς περιορισμούς.

Σκοπεύουμε να υλοποιήσουμε ένα γραφικό περιβάλλον για το χρήστη ο οποίος θα έχει τη δυνατότητα να αναζητεί λέξεις φράσεις που επιθυμεί. Η επιθυμητή συμπεριφορά που θέλουμε να επιτύχουμε είναι να εμφανίζεται σε αυτόν τα ‘Documents’ που θα περιλαμβάνουν τις λέξεις φράσεις κλειδιά που επιλέγει ο χρήστης. Επίσης θέλουμε να δίνουμε τη δυνατότητα στο χρήστη της αποκλειστικής αναζήτησης σε κάποια συγκεκριμένα πεδία . Για παράδειγμα να αναζητά λέξεις φράσεις που υπάρχουν σε συγκεκριμένα χαρακτηριστικά και όχι σε όλα. Δηλαδή θα προσπαθήσουμε να δημιουργήσουμε ευρετήρια για διάφορα ερωτήματα που θα μπορεί ο χρήστης να αναζητήσει. Θα επιδιώξουμε να κρατάμε ιστορικό τις φράσεις κλειδιά που αναζητεί ο χρήστης αλλά και τα αποτελέσματα των αναζητήσεων σε κάθε αναζήτηση για να προτείνουμε στο χρήστη έως ότου αναζητήσει ξανά τον συνδυασμό των προηγούμενων αναζητήσεων.

Για τη συλλογή έχουμε για παράδειγμα:

1η εγγραφή

s1,Movie,Dick Johnson Is Dead,Kirsten Johnson,,United States,"September 25, 2021",2020,PG-13,90 min,Documentaries,"As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable." **2η εγγραφη**

s2,TV Show,Blood & Water,,"Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng",South Africa,"September 24, 2021",2021,TV-MA,2 Seasons,"International TV Shows, TV Dramas, TV Mysteries","After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming star is her sister who was abducted at birth." **document 1:**

- show_id=s1
- Type=Movie
- Title=Dick Johnson Is Dead
- Director=Kirsten Johnson
- Cast=null

- Country=United States
- Date_added="September 25, 2021"
- Release_year=2020
- Rating=PG-13
- Duration=90 min
- Listen_in=Documentaries
- Description= "As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable."

Αντίστοιχα θα προκύψουν και τα υπόλοιπα μετά το parsing για το document 2 και ούτω κάθε εξής.

Στην αναζήτηση έστω ότι ο χρήστης θέλει να αναζητήσει την λέξη Johnson . Θέλουμε για αρχή να λέμε στο χρήστη ότι υπάρχει στο document 1 ενώ όσο το κάνουμε αυτό μπορούμε να έχουμε μία λίστα με τα πεδία που περιέχουν τη λέξη κλειδί . Εδώ δηλαδή να λέμε στο χρήστη αν θέλει να δει ακριβώς τι υπάρχει για αυτή την φράση κλειδί

Title=Dick Johnson Is Dead

Director=Kirsten Johnson

Ενώ στο ιστορικό μπορούμε να κρατάμε Johnson ,Dick Johnson Is Dead,Kirsten Johnson για να δούμε αν υπάρχει κάποια αντιστοιχία με κάποια νέα αναζήτηση για να προτείνουμε την εμφάνισή τους. Δηλαδή αν στην συνέχεια ο χρήστης αναζητεί την λέξη Dead να προτείνουμε άμεσα Dick Johnson Is Dead.

UpdateFinalVersion

Το πρόγραμμα αποτελείται από τις παρακάτω κλάσεις.

1. Corpus
2. IndexCreator
3. Search
4. AutoComplete
5. MovieCollectorGUI

1. Η κλάση **Corpus** είναι υπεύθυνη για την συλλογή των αρχείων από έναν κατάλογο στον προσωπικό μας Η/Υ. Δημιουργεί μια λίστα η οποία διατηρεί τα path των αρχείων που έχουμε. Στην περίπτωση μας, έχουμε μόνο ένα αρχείο .

2. Η κλάση **IndexCreator** υλοποιεί την δημιουργία των ευρετηρίων που θα αξιοποιήσουμε για την εύρεση των αναζητήσεων μας. Επειδή ακριβώς δουλέψαμε με μία έτοιμη csv λίστα με ταινίες και σειρές ήταν εύκολο με ένα αντικείμενο της κλάσης να δημιουργούμε Documents με τα πεδία `{ "type", "title", "director", "country", "rating", "description" };` ,χωρίς να απαιτείται να ξανά δημιουργούμε άλλο ευρετήριο για να κάνουμε parsing κάποιο άλλο πεδίο. (Δηλαδή με ένα διάβασμα όλου του αρχείου έχουμε όλα τα documents που παράγονται με τα πεδία τους, ανεξαρτήτως αν ο χρήστης έχει πειράξει την δημιουργία του ευρετηρίου με βάση κάποιο από τα παραπάνω πεδία που διατηρούμε.) Η κεντρική ιδέα του parsing είναι αυτή που περιεγράφηκε στην εισαγωγική αναφορά. Δλδ. Για κάθε εγγραφή στο csv αρχείο κρατάμε τα παραπάνω πεδία και δημιουργούμε Document με τα αντίστοιχα πεδία και στη συνέχεια προσθέτουμε αυτά τα Document στον IndexWriter. Επίσης , σε αυτόν προσθέτουμε και Document που σχετίζονται με το SearchHistory του χρήστη όταν χρειάζεται.

3. Η κλάση **Search** είναι υπεύθυνη για την αναζήτηση των λεκτικών μονάδων του χρήστη. Υλοποιεί τα quarrys με βάση την επιλογή του χρήστη. Οι επιλογές που έχει ο χρήστης είναι να αναζητά λέξεις κλειδιά σε ένα συγκεκριμένο πεδίο ή σε όλα τα πεδία που κρατάμε. Έτσι έχουμε δυο τύπους Parser :

    1. QueryParser: που ψάχνουμε τις λέξεις σε ένα συγκεκριμένο πεδίο που έχει επιλέξει ο χρήστης .

    2. MultiFieldQueryParser: που ψάχνουμε τις λέξεις σε όλα τα πεδία που κρατάμε.

Επίσης στην κλάση αυτή διαχειριζόμαστε και το Sort των αποτελεσμάτων. Βρήκαμε ότι η Lucene διαθέτει μόνη της 2 τύπους `RELEVANCE, INDEXORDER`, οπότε διαχειριζόμαστε την εκτύπωση με βάση την επιλογή του χρήστη ως προς αυτό. Επίσης, δίνουμε στο χρήστη να ταξινομήσει τα δεδομένα και ως προς τα πεδία `{"type", "title"}.`

4. Η κλάση **AutoComplete** είναι υπεύθυνη στο να προτείνει στο χρήστη με βάση προηγούμενες αναζητήσεις του Documents που ταιριάζουν σε αυτό που πληκτρολογεί. Όπως αναφέραμε στην κλάση **IndexCreator** δημιουργούμε documents που σχετίζονται με το SearchHistory. Αυτό που κάνουμε εδώ είναι να δημιουργούμε `FuzzyQuery` που σχετίζεται με το πεδίο που έχουμε δώσει στα Documents του ιστορικού. Έτσι αποθηκεύουμε σε μία λίστα το περιεχόμενο που υπάρχει στο ιστορικό που "μοιάζει" με αυτό που ήδη πληκτρολογεί ο χρήστης και του προτείνουμε αν θέλει να επιλέξει αυτό για αναζήτηση.

6. Η κλάση **MovieCollectorGUI** υλοποιεί την διεπαφή με το χρήστη. Περιλαμβάνει το γραφικό περιβάλλον που εμφανίζεται στο χρήστη και είναι υπεύθυνο να καλεί τις απαιτούμενες λειτουργίες με βάση τις επιλογές του χρήστη.

Για την ανάπτυξη του προγράμματος κάναμε χρήση του αρχείου **https://www.kaggle.com/datasets/shivamb/netflix-shows** με μία μικρή χειροκίνητη τροποποίηση. Παρατηρήσαμε ότι σε σε κάποιες γραμμές (2 ή 3 ήταν κάπου στο τέλος) υπήρχαν κάποιες newline ανάμεσα στα πεδία που μας δημιουργούσαν πρόβλημα με το parsing οπότε τις σβήσαμε χειροκίνητα.

Το αρχείο αυτό περιλαμβάνει 6131 Movies και 2676 Tv show. Αυτό το βρήκαμε με αναζήτηση απευθείας από το πρόγραμμά μας . πχ με type: Tv , type: Movie. Το αρχείο περιλαμβάνει συνολικά 8807 εγγραφές συνολικά , οπότε αν κάνουμε την πρόσθεση προκύπτει ότι λειτουργεί σωστά.

Λεπτομέρειες που αφορούν τα ερωτήματα και τον τρόπο λειτουργίας θα αναφερθούν στο βίντεο παρουσίαση.