# **ΜΥΥΕ003: Ανάκτηση Πληροφορίας**

# **Εαρινό Εξάμηνο 2021-2022**

**Εργασία: Μηχανή αναζήτησης ταινιών**

Σωτήρης Παναγιώτου, ΑΜ: 4456

Νικολέττα Μπιντζηλαίου, ΑΜ: 4435

* Link στο GitHub repository της εργασίας:

https://github.com/sotblad/anaktisi

* Read me:

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα(A) Τα δεδομένα συλλέχθηκαν από την σελίδα imdb με scraper. Συγκεκριμένα, συλλέξαμε τους τίτλους που εμφανίζονται με αύξοντα αριθμό δημοφιλίας- αριθμό ψήφων.

(<https://www.imdb.com/search/title/?title_type=feature,tv_movie,short>)

Από τη κάθε featured ταινία, συλλέγουμε μέσω του scraper τον τίτλο, την ημερομηνία που βγήκε, το είδος, την βαθμολογία της στα 100, την διάρκεια της σε λεπτά, την περιγραφή/περίληψη, τον/τους σκηνοθέτες και τους ηθοποιούς με πρωταγωνιστικούς ρόλους. Αυτές τις πληροφορίες τις αποθηκεύουμε σε ένα csv αρχείο, όπου οι παραπάνω κατηγορίες χωρίζονται σε στήλες με βάση το χαρακτήρα «|».

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Π.χ.

Παραθέτουμε στο τέλος σύντομο read\_me\_scraper για την λειτουργία του.

(Β)

Στόχος του συστήματος μας είναι ο χρήστης να μπορεί να ψάξει κάποια πληροφορία για μία ταινία και το σύστημα να του επιστρέφει τον τίτλο και μια σύντομη περιγραφή της πιο σχετικής (με το μεγαλύτερο score) και τους τίτλους από τις επόμενες 9 σχετικές αντίστοιχα.

Ανάλυση κειμένου και κατασκευή ευρετηρίου: Για την προεπεξεργασία των άρθρων για τη δημιουργία του εγγράφου, χρησιμοποιούμε τον StandardAnalyzer, που είναι ο default της Lucene. Δεν έχουμε πληροφορία για urls ή emails, αλλά θέλουμε να αφαιρούνται τα stop words και να γίνονται lowercase τα tokens.

Η μονάδα εγγράφου που θα προσθέτουμε στον IndexWriter, θα είναι μια ταινία σαν ‘σύνολο’, σαν αντικείμενο, δηλαδή μια ταινία με όλα τα χαρακτηριστικά της, μια γραμμή από το αρχείο csv. Ο csv parser που διαβάζει το csv, παίρνει κάθε στήλη του εγγράφου ξεχωριστά και την αποθηκεύει σε μία λίστα getColumns. Φτιάχνουμε ένα entity Movie, με attributes title, year, genre, rating, duration, description, directors, stars. Τα πεδία year, rating, duration θα είναι τύπου int και τα title, description String. Τα πεδία genre, directors και stars αρχικά είναι λίστες από String, μιας που μπορούν να περιέχουν πάνω από μία ‘τιμή’. Μέσα στο Movie, τα μετατρέπουμε σε ένα πολύ μεγάλο String. Έτσι, κάθε μονάδα εγγράφου θα αποτελείται από τα 8 πεδία (fields) αντίστοιχα, τα title, year, genre, rating, duration, description, directors, stars όπως αναφέρθηκαν παραπάνω, για να μπορεί να τα διαχειριστεί η Lucene.

Αναζήτηση: Για την αναζήτηση, θα δημιουργήσουμε QueryParser που θα χρησιμοποιεί επίσης τον StandardAnalyzer, για την επεξεργασία του input του χρήστη. Θα δημιουργήσουμε IndexSearcher που θα παίρνει το query και θα ψάχνει τα έγγραφα. Ανάλογα με το αν ο χρήστης θέλει να ψάξει σε όλο το λεξικό ή μόνο σε ένα συγκεκριμένο πεδίο, θα πρέπει να διαχειριζόμαστε το query διαφορετικά.

Όσον αφορά την ταξινόμηση, μας ενδιαφέρει το default sorting mode του scoreDocs, ώστε να εμφανίζονται ανάλογα με το πόσο σχετικά είναι.

Επίσης, θα μας ενδιέφερε ο χρήστης να μπορεί να κάνει και κάποια εξειδικευμένη αναζήτηση συνδυάζοντας λέξεις κλειδιά, π.χ. ψάχνοντας ταινίες που ανήκουν και στην κατηγορία ‘drama’ και ‘mystery’. Για αυτή την υλοποίηση θα χρειαστούμε BooleanQuery.

Tέλος, θα μας ενδιέφερε και η υλοποίηση κάποιου FuzzyQuery για να ‘διορθώνουμε’ ορθογραφικά λάθη του χρήστη σε τυχόν λέξεις κλειδιά και να βγαίνουν σχετικά αποτελέσματα.

Παρουσίαση Αποτελεσμάτων: Θα φτιάξουμε ένα interface ώστε να μπορεί να εισάγει με ευκολία ο χρήστης την λέξη κλειδί σε ένα search bar και να εμφανίζονται τα 10 πρώτα αποτελέσματα σε ένα πίνακα με τονισμένες τις λέξεις κλειδιά και δυνατότητα να προχωράει στα επόμενα. Ιδανικά θα πρέπει να μπορεί να επιλέξει αν θα κάνει αναζήτηση σε όλο το λεξικό ή αν θα ψάξει πληροφορία σε ένα μόνο πεδίο που επιλέγει, πατώντας κάποιο custom search κουμπί. Επιπρόσθετα, με κάποιο κουμπί θα μπορούμε να ταξινομούμε τα αποτελέσματα που έχουν βγει ανάλογα με το πεδίο που θέλει ο χρήστης.

* Scraper

Τρέχουμε το αρχείο imdbScraper.py δίνοντας ως όρισμα τον αριθμό των ταινιών που θέλουμε να αποθηκεύσουμε. Ο αριθμός αυτός χρησιμοποιείται για να ξέρουμε πόσες σελίδες από τη imdb ιστοσελίδα θα χρειαστεί να «ψάξουμε». Φτιάχνουμε το αρχείο scraped\_movies.csv στο οποίο γράφουμε ώστε να έχει τις στήλες-πεδία: title|year|genre|rating|duration|description|directors|starsList. Επίσης, φτιάχνουμε μία άδεια τελική λίστα με ταινίες.

Όσο δεν έχουμε φτάσει των αριθμό σελίδων που χρειαζόμαστε:

• Το url των σελίδων που θα χρειαστούμε μέχρι τις πρώτες 10.000 ταινίες είναι το "https://www.imdb.com/search/title/?title\_type=feature,tv\_movie,short&count=250&start=" + str(1+(i\*250)), όπου i=0..αριθμό των συνολικών σελίδων που θα χρειαστούμε. Μετά απο αυτές, το url αλλάζει (Π.χ.https://www.imdb.com/search/title/?title\_type=feature,tv\_movie,short&count=250&after=WzE0MTI0LCJ0dDE3MzIxMjMwIiw5NzUxXQ%3D%3D ) και έτσι λαμβάνουμε το url της επόμενης σελίδας κατευθείαν απο τον υπερσύνδεσμο στην html.

• Αν ο αριθμός των ταινιών που εισάγαμε στο terminal είναι μικρότερος των 250, τότε ζητάμε αυτό τον αριθμό featured, διαφορετικά, αφαιρούμε από τον συνολικό αριθμό τις πρώτες 250 υπάρχουσες και ζητάμε ακριβώς 250.

• Ανοίγουμε το url και κόβουμε τον html κώδικα από το '<div class="lister-list">' μέχρι το '\div class="desc">', που περιέχει τις πληροφορίες που μας ενδιαφέρουν για όλες τις ταινίες. Ανάλογα με τη σελίδα στην οποία είμαστε βρίσκουμε και κρατάμε το κατάλληλο url για την επόμενη. Καλούμε την την μέθοδο twofifty με ορίσματα τις πληροφορίες που μας ενδιαφέρουν στην html, την λίστα με τις ταινίες και των αριθμών των ζητούμενων featured, για να ξεχωρίσουμε και να αποθηκεύσουμε τις ταινίες με τα πεδία τους.

• def twofifty(movies, moviesList, want):

Χρησιμοποιούμε έναν μετρητή για να μετράμε τις ταινίες που είναι "έτοιμες": για τις οποίες έχουμε αποθηκεύσει όλα τα απαραίτητα πεδία. Όσο αυτός ο μετρητής είναι μικρότερος από των αριθμών των ζητούμενων featured, «σπάμε» την πληροφορία (movies) σε ξεχωριστές ταινίες, σε κάθε μία από τις 250 featured. Με τη σειρά:

1. Κρατάμε το header και από αυτόν τον τίτλο
2. Κρατάμε το κατάλληλο κομμάτι του header για να βρούμε τη χρονολογία. Αν ανοίγει παρένθεση και ο τίτλος συνεχίζει, την προσθέτουμε στο πεδίο title, διαφορετικά, η χρονολογία έχει πάντα 4 ψηφία.

Στη συνέχεια:

1. Ψάχνουμε το επόμενο κομμάτι που μας ενδιαφέρει που έχει να κάνει με το info για τη διάρκεια, το είδος και την βαθμολόγηση. Μία ταινία μπορεί να ανήκει σε πάνω από ένα είδος και για αυτό είναι λίστα.
2. Βρίσκουμε το κομμάτι για την περίληψη-περιγραφή. Μπορεί να «λείπει» ή να μην είναι ολοκληρωμένη λόγω αναλυτικής περιγραφής.
3. Προχωράμε στο κομμάτι για το σκηνοθέτη και τους ηθοποιούς. Οι σκηνοθέτες μπορεί να είναι πάνω από ένας και οι ηθοποιοί είναι πολλαπλοί συνηθώς, οπότε πρέπει να «χωριστούν» αντίστοιχα και να μπουν μέσα σε μία λίστα για την κάθε ταινία.

Μπορούμε πλέον να προχωρήσουμε στην επόμενη featured. Αυξάνουμε τον μετρητή και προσθέτουμε τα «πεδία» μας σε μία προσωρινή λίστα. Ελέγχουμε ότι η λίστα αυτή δεν υπάρχει ήδη στη συνολική λίστα με τις ταινίες και την προσθέτουμε, γράφοντας τις αντίστοιχες πληροφορίες στο csv αρχείο μας.