Εργασια Ανακτηση Πληροφοριας 2022

Μάριος Ζίχναλης 3226

https://github.com/marioszih/Data-Retrieval-2022

Πρόλογος

Η παρακάτω αναφορά αποτελεί την ανάλυση των βημάτων που πραγματοποιήθηκαν για το σχεδιασμό του συστήματος καθώς και την ανάπτυξη μιας απλής μηχανής αναζήτησης. Η αναφορά περιέχει τον τρόπο συλλογής των εγγράφων που χρειάστηκαν ώστε να υπάρχει μία βάση δεδομένων, την ανάλυση κειμένου και τα ευρετήρια που θα δημιουργηθούν, την αναζήτηση των αποτελεσμάτων, κυρίως με λέξεις κλειδιά και τέλος την παρουσίαση τους με βάση τη συνάφεια που έχουν με το ερώτημα. Στον φάκελο στο github ανέβασα τα jars που χρησιμοποιεί η εργασία, καθώς και τον source code.

Επιλογή αρχείου ταινιών

Το αρχείο που επιλέχθηκε είναι αυτο που είχε αναφερθεί και στο πρώτο συνοπτικό report για την εργασία. Είχε 16.000 εγγραφές εκ των οποίων, όλες αποτελούν ταινίες. Έπειτα απο την προεπεξεργασία του και την αφαίρεση null τιμών, έμειναν 7600 εγγραφές. Ο λόγος επιλογής του συγγεκριμένου αρχείου ειναι αφενός μεν το γεγονός οτι περιέχει αρκετές εγγραφές για να εχω ενα μεγάλο δείγμα αναζητήσεων, αφετέρου δε, περιέχει όλα τα απαραίτητα πεδία για να γίνεται μια σωστή περιγραφή μιας ταινίας και να διευκολύνει την αναζήτηση της, καθώς και κάποια έξτρα τα οποία θα μπορούσαν να χρησιμοποιηθούν για καλύτερη κατηγοριοποίηση των ταινιών. Στην συγκεκριμένη υλοποίηση κράτησα μόνο αυτα που θεώρησα εγώ απαραίτητα, και θα ήθελα να περιέχει η μηχανή αναζήτησης που θα χρησιμοποιούσα. Απο τις 23 στήλες κράτησα τις 12 εκ των οποίων οι 11 χρησιμοποιούνται για την αναζήτηση και η 12^{η} είναι το link για την σελίδα της ταινίας στο rotten tomatoes. Τα πεδία που κρατήθηκαν ειναι τα εξής:

- Rotten Tomatoes Link
- Movie Title
- Movie Info
- Critics Consensus
- Rating
- Genre
- Directors
- Writers
- Cast
- Runtime
- Tomatometer Rating
- Audience Rating

Η προ-επεξεργασία έγινε με ενα python script το οποίο περιέχεται στο github με το όνομα prep.py. Το τελικό αρχείο το οποίο χρησιμοποιεί η εφαρμογή ειναι το edited_file.csv.

Ευρετηριοποίση

Αναφορικά με την ανάλυση κειμένου, η πρώτη γραμμή του αρχείου περιέχει συγκεκριμένα πεδία που ελαχιστοποιούν την περιεκτικότητα πληροφορίας για το εκάστοτε άρθρο. Στη συνέχεια δημιουργείται το index το οποίο περιέχει docs όπου κάθε doc έχει fields. Η προεπεξεργασία των άρθρων για τη δημιουργία του εγγράφου γίνεται με τη βοήθεια του **Standard Analyzer** ή αλλιώς default analyzer. Ο αναλυτής διαχωρίζει ,το κείμενο σε λέξεις, τα γράμματα από τα σύμβολα της γλώσσας και αφαιρεί τις τερματικές λέξεις (stop words). Η επιλογή του συγκεκριμένου αναλυτή έγινε λόγω της δομής του υπάρχοντος εγγράφου. Η μονάδα εγγράφου είναι ουσιαστικά η γραμμή του αρχείου (csv), η πληροφορία για ένα συγκεκριμένο άρθρο. Επιπρόσθετα, για την προσπέλαση των προαναφερθέντων πεδίων αποτελεί ανάγκη η κατασκευή βοηθητικών δομών, ευρετηρίων ώστε να μειωθεί ο χρόνος αναζήτησης και ο χρήστης να έχει έγκαιρα τα αποτελέσματα που επιθυμεί να έχει πρόσβαση. Για την συγκεκριμένη εφαρμογή, θα χρησιμοποιηθούν αντεστραμμένα ευρετήρια (inverted indexes) για τα παρακάτω πεδία: **title, summary, critics, cast, director, audience score, tomato score, writers, genre, runtime, rating**.

Αναζήτηση ταινιών

Η ελάχιστη απαίτηση ήταν η αναζήτηση άρθρων με λέξεις κλειδία, υλοποιήθηκε και η αναζήτηση χαρακτηριστικών όρων σε συγκεκριμένα πεδία, κάνοντας την αναζήτηση πάνω στο αντεστραμμένο ευρετήριο του συγκεκριμένου πεδίου. Πιο συγκεκριμένα η αναζήτηση χρησιμοποιεί 2 ειδών parsers. Πρώτα έχουμε τον MultiFieldQueryParser όπου χρησιμοποιήται αν ο χρήστης δεν διαλέξει κάποιο συγκεκριμένο πεδίο αναζήτησης, οπότε η αναζήτηση γίνεται σε όλα τα πεδία. Ο δεύτερος ειναι ο QueryParser, ο οποίος είναι ουσιαστικά ένας διερμηνέας που μας επιτρέπει να κάνουμε ερωτήσεις πάνω σε συγκεκριμένο πεδίο δίνοντάς του ένα αλφαριθμητικό. Έχει φτιαχτεί και η λειτουργία ιστορικού αναζητήσεων. Στο πρώτο παράθυρο που ανοίγει όταν τρέξει η εφαρμογή, ο χρήστης μπορεί να διαλέξει αντί να κάνει μια νέα αναζήτηση, να ανοίξει το ιστορικό αναζητήσεων και να ξανα κάνει μια αναζήτηση που είχε κάνει και πριν. Η πληροφορία αυτή διατηρητε σε ένα txt αρχείο, λειτουργεί μόνο για την ώρα που η εφαρμογή εκτελέιται και δεν διατηρείτε για επόμενες εκτελέσεις. Δεν μπόρεσα να υλοποιήσω την λειτουργία πρότασης εναλλακτικών ερωτημάτων ανάλογα με της άλλες αναζητήσεις. Επίσης δεν κατάφερα να κανώ δυνατή την αναζήτηση range, δηλαδή για παράδειγμα να ψάχνει ταινίες με runtime 100 λεπτά και παραπάνω η ταινίες με audience rating παραπάνω απο 77%.

Παρουσίαση Αποτελεσμάτων

Η παρουσίαση των αποτελεσμάτων γίνεται μέσω της χρήσης ενός ξεχωρηστού frame της εφαρμογής με το όνομα ResultFrame. Εκεί τα αποτελέσματα παρουσιάζονται ανα 10, (με την δυνατότητα αλλαγής αυτού του αριθμού), και ο χρήστης μπορεί με την χρήση των 2 κουμπιών στο πάνω και κάτω μέρος του frame να αλλάζει σελίδες, όπως ζητούσε η άσκηση. Η παρουσίαση εξαρτάται απο το τί θα αναζητήση ο χρήστης. Αν ο χρήστης επιλέξει να αναζητήσει στο πεδίο summary τότε η εφαρμογή θα εμφανίσει την λέξη που αναζητήθηκε απο τον χρήστη, highlighted σε μόνο στο πεδίο το οποίο επέλεξε να κάνει ο χρήστης την αναζήτηση. Για παράδειγμα αν ψάξω με βάση τίτλο ταινίας την λέξη good, ταινίες που περιέχουν την λέξη good στο summary η στο critic wrote δεν θα έχουν την λέξη good highlighted, ενώ αν ψάξει την λέξη good στο πεδίο summary, τότε θα γίνει highlight μόνο στο πεδίο summary. Για το highlighting χρησιμοποιήθηκε ο έτοιμος highlighter της lucene. Πέρα απο το highlighting, χρησιμοποιήθηκε η μέθοδος hyperlinkMaker. Εκεί αυτο που γίνεται ειναι οτι κάνω τον τίτλο της ταινίας ένα clickable label το οποίο με το που πατηθεί, τότε κανει search στον τοπικό browser του συστήματος, το link της ταινίας στο rotten tomatoes και την παρουσιάζει στο site. Η εμφάνιση των αποτελεσμάτων γίνεται σε ομάδες των 10. Στο MainWindow υπάρχει ενα κουμπί το οποίο, όταν πατηθεί ανοίγει ενα μικρό παράθυρο στο οποίο ο χρήστης πληκτρολογεί έναν αριθμό μεταξύ του 1 και του 10 και πατώντας Apply αλλάζει τον αριθμό αποτελεσμάτων που φαίνονται στο παράθυρο. Στο ResultWindow πάνω και κάτω υπάρχουν 2 κουμπία με τα ονόματα next page και previous page. Αυτά όπως φαίνεται απο το όνομα τους αλλάζουν το ResultFrame για να εμφανιστούν τα επόμενα 10 (η όσα επιλέξει ο χρήστης) αποτελέσματα. Για να γίνει η αλλαγή του αριθμού αποτελεσμάτων πρέπει ο χρήστης να το αλλάξει πριν κάνει εκ νέου search. Η αλλαγή παραμένει και όταν ο χρήστης κάνει αναζήτηση και μέσω ιστορικού. Το default result size είναι 10. Δεν κατάφερα να κάνω κάποιο διαφορετικό τρόπο ομαδοποίησης.

Χρήσιμα Link

Το άρχειο βρεθηκε στο Kaggle. Ο συνδεσμος στον οποιο βρηκά το άρχειο ειναι ο εξης:

https://www.kaggle.com/datasets/heyueyuan/rottentomatoesmoviesandcriticsdatasets

Εδω έχουμε την σελίδα για την μέθοδο HyperlinkMaker:

https://www.codejava.net/java-se/swing/how-to-create-hyperlink-with-jlabel-in-java-swing

Εδω η σελίδα για τον highlighter της lucene:

https://www.bitspedia.com/2015/07/lucene-highlighter-tutorial-with-example.html

Εδω το introduction που διάβασα για την lucene:

https://www.baeldung.com/lucene

Επίλογος

Στην συγκεκριμένη αναφορά περιγράφεται πως περίπου λειτουργεί η εφαρμογή και ο τρόπος με τον οποίο υλοποιήθηκαν οι λειτουργίες της. Για το πως χρησιμοποιήτε δείτε το demo video που ανέβασα στο github.