Αναζήτηση του ArchWiki με Lucene

Παπασταθοπουλος Κωστας
Τμήμα μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων
Διεθνές Πανεπιστήμιο Ελλάδας
Θεσσαλονικη, Ελλαδα
papastathopoulosko@gmail.com

***Περίληψη*---Το παρόν έγγραφο αποτελεί ενα project report που παρουσιάζει τον σχεδιασμό και την υλοποίηση ενός συστήματος ευρετηρίασης και αναζήτησης του ArchWiki με τη χρήση του Apache Lucene. Οι βασικές λειτουργίες του backend της εφαρμογής περιλαμβάνουν τη δημιουργία ενός τοπικού ευρετηρίου για το wiki, την αναζήτησή του και την έκθεση της σελιδοποίησης για τους πελάτες μεταξύ άλλων τροποποιήσιμων ιδιοτήτων. Το UI είναι χτισμένο με δυναμικά στοιχεία UI, υποστηρίζει σελιδοποίηση, εναλλαγή θεμάτων και μπορεί να χρησιμοποιήσει ορισμένες από τις λειτουργίες που εκτίθενται από το backend module. Οι προκλήσεις που αντιμετωπίζονται περιλαμβάνουν το modularity και τον συγχρονισμό καταστάσεων, ενώ οι περιορισμοί περιλαμβάνουν την υποστήριξη μονάχα μίας γλώσσας και τον χειρισμό της πολυπλοκότητας μερικών ερωτημάτων. Οι μελλοντικές εργασίες αποσκοπούν στην ενίσχυση της επεκτασιμότητας, της υποστήριξης πολλαπλών γλωσσών και την βελτιστοποίηση εμπειρίας χρήστη. Το project επιδεικνύει μια συντηρήσιμη, αρθρωτή προσέγγιση για την ενσωμάτωση αποδοτικών αλγορίθμων αναζήτησης με ένα σύγχρονο UI**

***Index Terms*---Lucene, Kotlin, Compose, Ktor, Jsoup, StandfordCoreNLP**

I. Εισαγωγή

Έχουν δημιουργηθεί δύο modules που είναι υπεύθυνα για ανεξάρτητες εργασίες, ώστε να γίνει ο κώδικας πιο οργανωμένος και να βελτιωθεί η συντηρησιμότητα για μελλοντικές εργασίες, ενώ παράλληλα προσφέρεται διαχωρισμός προβλημάτων. Τα modules που δημιουργήθηκαν είναι τα 'searchengine' και 'searchengineui'. Το 'searchengine' επικεντρώνεται στη δημιουργία ενός ευρετηρίου και στην εκτέλεση αναζητήσεων σε αυτό, αποτελώντας το backend του συστήματος. Το 'searchengineui' εξαρτάται και αλληλεπιδρά με το 'searchengine' για να παρέχει μια φιλική προς το χρήστη διεπαφή για την αλληλεπίδραση με αυτό.

II. Module SearchEngine

Το Module `SearchEngine` είναι μια βιβλιοθήκη που είναι υπεύθυνη για τον χειρισμό των backend διαδικασιών ανάκτησης, ευρετηρίασης, αναζήτησης και κατάταξης εγγράφων. Εκμεταλλεύεται πολλαπλές βιβλιοθήκες, όπως η Ktor για τη λήψη των html σελίδων του wiki, Jsoup για την ανάλυση των σελίδων HTML και την αφαίρεση των περιττών ετικετών και χαρακτηριστικών StanfordCoreNLP για τη λημματοποίηση και, τέλος, το Lucene του Apache για την επεξεργασία, την ευρετηρίαση και την αναζήτηση. Περιέχει τέσσερις κλάσεις που θα αναλυθούν περαιτέρω στις επόμενες ενότητες.

Α΄. Κλάση SearchEngine

Αυτή η κλάση χρησιμεύει ως Πρόσοψη, παρέχοντας μια ενιαία διεπαφή για τις σύνθετες λειτουργίες που υποστηρίζονται απο την βιβλιοθηκη . Ενθυλακώνει τις εσωτερικές λεπτομέρειες υλοποίησης και εκθέτει ένα API για το 'searchengineui' module, τηρώντας τις αρχές της αρθρωτότητας και του διαχωρισμού των προβλημάτων. Όταν αρχικοποιείται, ελέγχει αν υπάρχει ο φακελος που περιέχει το index (η προεπιλογή είναι cwd/data/site/ αν δεν έχει δοθεί) και αν όχι, κατεβάζει το πακέτο arch-wiki-docs χρησιμοποιώντας το ktor. Στη συνέχεια χρησιμοποιεί το zstd-jni του luben για να αποσυμπιέσει το τμήμα του αρχείου που ειναι σε Zstandard (`.zst`) μορφη και τέλος τη βιβλιοθήκη apache comms compressions για να εξάγει το τμήμα ('.tar') του αρχείου. Τέλος, φιλτράρει τα αποτελέσματα και κρατάει μόνο τον φακελο που περιέχει τις αγγλικές σελίδες, τις εικόνες και το μοναδικό αρχείο css που υπάρχει, έτσι ώστε το άνοιγμα μιας σελίδας να έχει την ίδια διαμόρφωση με την online έκδοση του wiki. Στην περίπτωση που ο κατάλογος υπήρχε αυτά τα βήματα παραλείπονται εντελώς.

Β΄. Κλάση Lucene

Αυτή η κλάση αντιπροσωπεύει τη βασική λογική του module, υλοποιώντας το domain logic, όπως η αναζήτηση και η ευρετηρίαση. Είναι το κύριο συστατικό που είναι υπεύθυνο για την εκτέλεση των κρίσιμων λειτουργιών του module, τηρώντας τις αρχές του διαχωρισμού ανησυχιών και της ενθυλάκωσης. Με την απομόνωση αυτής της λειτουργικότητας, το σύστημα εξασφαλίζει υψηλή επαναχρησιμοποίηση, δυνατότητα ελέγχου και συντηρησιμότητα. Διαθέτει δύο functions, το createIndex και το searchIndex.

- **createIndex :** Χρησιμοποιεί το EnglishAnalyzer για την επεξεργασία της αγγλικής γλώσσας (stop words) και τη μετατροπή της εισόδου σε tokens. Ελέγχει το υπάρχοντα δείκτη για να αποφύγει την εκ νέου δεικτοδότηση αρχείων με την αναζήτηση της διαδρομής καθε αρχείου. Δημιουργεί και ενημερώνει το ευρετήριο Lucene, ενώ παρέχει ζωντανές ενημερώσεις προόδου καλώντας την παράμετρο lambda με τις ενημερωμένες τιμές. Η επεξεργασία του αρχείου (html, lematization) γίνεται ασύγχρονα με πολλαπλά threads χρησιμοποιώντας Kotlin coroutines για να επιταχύνει τη δημιουργία του ευρετηρίου.
- **searchIndex** Αυτή η συνάρτηση δημιουργεί ερωτήματα Lucene με βάση τα δεδομένα που εισάγει ο χρήστης συν-

δυάζοντας QueryParser, PhraseQueries, TermQueries, PrefixQueries και FuzzyQueries για την αντιστοίχιση των δεδομένων που εισάγει ο χρήστης. Τα ερωτήματα τυλίγονται σε BoostQueries για να δοθεί σε κάθε ένα από τα ερωτήματα διαφορετική βαρύτητα και να βελτιωθούν τα αποτελέσματα αναζήτησης. Προσπαθεί επίσης να ταιριάξει αποσπάσματα του ερωτήματος για κάθε έγγραφο και να επιστρέψει μια συμβολοσειρά για επισήμανση με τη χρήση του Lucene Highlighter, σε περίπτωση αποτυχίας επιστρέφεται μια standard συμβολοσειρά. Τα αποτελέσματα μπορούν να ταξινομηθούν με βάση την ημερομηνία τροποποίησης περνώντας true στην παράμετρο sortByDate. Οι χρήστες της βιβλιοθήκης μπορούν να ορίσουν προσαρμοσμένους περιορισμούς αποτελεσμάτων περνώντας μια τιμή στην παράμετρο resultsPerPage (το προεπιλεγμένο όριο είναι 10) για να αυξηθουν η να μειοθουν τα αποτελεσμάτων αναζήτησης. Υποστηρίζει αναζήτηση με BM25 (προεπιλογή) ή TF-IDF μέσω της παραμετρου searchWithTfIdfOnly .

*Γ΄. **Κλάση Lemmatizer***

Αυτή η κλάση κατέχει ένα μοναδικό pipe object με προσαρμοσμένες ιδιότητες για την επεξεργασία κειμένου και εκθέτει μια μέθοδο lemmatize η οποία λαμβάνει μια είσοδο και εκτελεί λημματοποίηση με τις προσαρμοσμένες ιδιότητες. Χρησιμοποιείται από την κλάση HTMLparser για να εκτελέσει προ επεξεργασία κειμένου μετά την αφαίρεση των ετικετών HTML και από τη λειτουργία Search της κλάσης Lucene για να βεβαιωθεί ότι εφαρμόζουμε τους ίδιους μετασχηματισμούς λέξεων στο ευρετήριο και στις αναζητήσεις.

*Δ΄. **Κλάση HTMLParser***

Αυτή η κλάση, κατά την αρχικοποίηση, δέχεται ως παράμετρο και αποθηκεύει ένα lambda function το οποίο δέχεται μια συμβολοσειρά και επιστρέφει μια επεξεργασμένη συμβολοσειρά. Αυτή η προσέγγιση επιτρέπει τη δυναμική διαχείριση του text processing, παρέχοντας ευελιξία ώστε σε μελλοντικές χρησης να μπορει να αντικατασταθει ευκολα η προ επεξεργασια κειμενου με διαφορετικη υλοποίηση. Η κλάση εκθέτει μια συνάρτηση parseHtmlToDocument, η οποία λαμβάνει ως όρισμα ένα αρχείο. Χρησιμοποιεί το Jsoup για να αφαιρέσει περιττές ετικέτες που δεν πρέπει να ευρετηριάζονται, όπως navigation, scripts κ.λπ. Στη συνέχεια, εκτελεί το αποθηκευμένο lambda function στο περιεχόμενο του εγγράφου και επιστρέφει ένα έγγραφο Lucene, το οποίο περιέχει τα ακόλουθα πεδία.

- **Τίτλος** Το HTML στοιχείο title αλλά χωρίς το τμήμα « -ArchWiki» στο τέλος.
- **Path** Το Path αρχείου όπου είναι αποθηκευμένο στον δυσκο.
- **Χρόνος τελευταίας τροποποίησης σε String** Ο χρόνος τελευταίας τροποποίησης σε μορφή String, ώστε να μπορεί εύκολα να εμφανιστεί στο UI.
- **Χρόνος τελευταίας τροποποίησης σε Long** Ο χρόνος τελευταίας τροποποίησης σε Long μορφή ώστε να μπορεί να χρησιμοποιηθεί για την ταξινόμηση.
- **Περιεχόμενο εγγράφου** Όλο το σημαντικό κείμενο του εγγράφου.

*Ε΄. **Wrapper Κλάσεις***

Υπάρχουν δύο wrapping κλάσεις στο Module και χρησιμοποιούνται και οι δύο για να εκθέσουν πληροφορίες εκτός του Module.

- **Κλαση WikiDocumentResult** Ένα object που αντιπροσωπεύει ένα έγγραφο το οποιο ταιριάζει με την αναζήτηση και περιέχει δεδομένα που θα εμφανιστούν στο χρήστη, με δεδομένα όπως τίτλος, βαθμολογία κ.λπ.
- **Κλαση LuceneWrapper** Ένα object που περιέχει μια λίστα με κάθε αντιστοιχία **WikiDocumentResult**, μαζί με τον αριθμό των συνολικών αντιστοιχιών που βρέθηκαν και τον αριθμό των συνολικών σελίδων.

III. Module SearchEngineUI

Αυτο το Module είναι υπεύθυνο για τον χειρισμό της αλληλεπίδρασης με το χρήστη παρεχοντας ενα GUI. Έχει εξάρτηση από το SearchEngine Module για να κάνει compile και να χρησιμοποιηθεί. Χρησιμοποιεί το Compose Multiplatform για την κατασκευή του UI μαζί με το Material 3 και Material Icons. Το σημείο εισόδου και τα τρία composable functions που ενεργούν σαν οθόνες σε αυτό το module θα αναλυθούν περαιτέρω παρακάτω.

Α΄. Main

Η είσοδος για το ui είναι το composable function 'main'. Είναι υπεύθυνο για το παρατήρηση του state της βιβλιοθήκης searchengine και την εμφάνιση της κατάλληλης οθόνης. Κατά τη δημιουργία της αρχικοποιεί την κλάση SearchEngine και την αποθηκεύει για όλες τις επανασυνθέσεις ώστε να μην χάνονται δεδομένα και να μην υποβαθμίζεται η απόδοση, αρχικοποιεί επίσης άλλες μεταβλητές που είναι υπεύθυνες για την κατάσταση της εφαρμογής. Τιμές όπως isDarkMode isIndexing κλπ. Οι μεταβάσεις κάθε οθόνης αντιμετωπίζονται με το AnimatedVisibility για να υπαρχουν animations κατα την μετάβαση των οθονων και να δημιουργείτε μια ευχάριστη εμπειρία χρήστη.

Β΄. DownloadScreen Composable

Αυτο το composable function καλείται κατά την ανάκτηση και εξαγωγή δεδομένων. Έχει μια εικονική κυκλική ένδειξη προόδου που δεν αλλάζει με βάση το ποσοστό της ενέργειας, αφού η ενέργεια δεν θα διαρκεί περισσότερο από μερικά δευτερόλεπτα.

Γ΄. IndexScreen Composable

Αυτή η συνάρτηση λαμβάνει δύο ορίσματα, ένα string για την εμφάνιση του στοιχείου που επεξεργάζεται εκεινη τη στιγμή και ένα float που αναπαριστά την πρόοδο σε μια κλίμακα από 0 έως 1.

Δ΄. SearchScreen Composable

Αυτό το function είναι αυτό με το οποίο ο χρήστης θα αλληλεπιδράσει περισσότερο. Διαθέτει μια γραμμή αναζήτησης που ελέγχει για ενημερώσεις και για κάθε ενημέρωση εκτελεί εκ νέου μια αναζήτηση. Διαθέτει ένα κουμπί για την ενεργοποίηση και απενεργοποίηση του σκοτεινού θέματος, ένα κουμπί δημιουργίας ευρετηρίου για την ενεργοποίηση της δημιουργίας του ευρετηρίου και ένα εικονίδιο με γρανάζι που αλλαζει την ορατότητα των πιο προηγμένων επιλογών. Οι πιο προηγμένες επιλογές είναι, «αναζήτηση μόνο για τίτλο» και «αναζήτηση με tf-idf», οι οποίες είναι και οι δύο απενεργοποιημένες στην προεπιλογή.

IV. Μελλοντικές εργασίες

- Πρόσθεση προηγμένων επιλογών φίλτρων (π.χ. εύρος ημερομηνίας, μέγεθος αρχείου) για να βελτιωθουν η επιλογές αναζήτησης.
- Πρόσθεση autocompletion και ιστορικό ερωτημάτων στη γραμμή αναζήτησης.
- Υποστήριξη πολλαπλών γλωσσών για λημματοποίηση, ευρετηρίαση και αναζήτηση.
- Να γίνει η ανάλυση και επεξεργασια πιο αφηρημένη για την χρήση του library με άλλα wiki.
- Δημιουργία περισσοτέρων test για την αναζήτηση, προς το παρόν γίνεται testing μόνο στα πιο βασικά κομμάτια της εφαρμογής.

Αναφορές

[1] Kastik, “GitHub - kastik/ComposeSearchEngine,” *GitHub*. Available: https://github.com/kastik/ComposeSearchEngine

[2] Kastik, “ComposeSearchEngine/searchenginedocs/src/ai.log at master · kastik/ComposeSearchEngine,” *GitHub*. Available: https://github.com/kastik/ComposeSearchEngine/blob/master/searchenginedocs/src/ai.log

[3] “arch-wiki-docs.” Available: https://archlinux.org/packages/extra/any/arch-wiki-docs/