ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ

‘Μηχανή αναζήτησης κριτικών χρηστών’

2η Φάση: Περιγραφή αρχικού σχεδιασμού

Μέλη Ομάδας

Γκαβαρδίνας Όθωνας, ΑΜ 2620
Μπουρλή Στυλιανή, ΑΜ 2774

Περιγραφή

1. Για κάθε εστιατόριο που έχουμε επιλέξει θα δημιουργήσουμε ένα έγγραφο, το οποίο θα περιλαμβάνει τα εξής πεδία, που θα αφορούν το εστιατόριο: όνομα, διεύθυνση, πόλη, κράτος, πλήθος αστεριών και κριτικές. Στη συνέχεια, θα χρησιμοποιήσουμε Index για να εισάγουμε το έγγραφο στο IndexWriter.
Για τη λειτουργικότητα 1, θα φτιάξουμε ένα αντεστραμμένο ευρετήριο που θα αφορά το πεδίο πόλη, έτσι ώστε να μπορεί να γίνει αναζήτηση με βάση τη γεωγραφική θέση του εστιατορίου. Επιπλέον, θα φτιάξουμε ένα αντεστραμμένο ευρετήριο που θα αφορά το πεδίο κριτικές, δηλαδή αφού διαχωρίσουμε το κείμενο των κριτικών σε tokens μέσω του Analyze θα δημιουργήσουμε το ευρετήριο.
Για τη λειτουργικότητα 2, θα φτιάξουμε ένα αντεστραμμένο ευρετήριο που θα αφορά το πεδίο όνομα εστιατορίου και ένα πεδίο για τις λέξεις κλειδιά της κάθε κριτικής.

2. Η διεπαφή που θα χρησιμοποιήσουμε θα αποτελείται από παράθυρα. Αρχικά, θα εμφανίζεται ένα παράθυρο, τύπου μενού, που θα διαθέτει επιλογή για αναζήτηση εστιατορίων και για αναζήτηση κριτικών. Ανάλογα με την αναζήτηση που θέλει να πραγματοποιήσει ο χρήστης, θα εμφανίζεται ένα παράθυρο με τους τρόπους που διατίθενται για κάθε επιλογή, τους τρόπους που διατίθενται για εμφάνιση των αποτελεσμάτων, και ένα πλαίσιο για εισαγωγή κειμένου. Για παράδειγμα, αν ο χρήστης θέλει να αναζητήσει εστιατόρια που βρίσκονται στην πόλη Α, θα επιλέγει αρχικά “Αναζήτηση εστιατορίου”, στη συνέχεια θα επιλέγει “Αναζήτηση με βάση το όνομα”, θα επιλέγει τρόπο εμφάνισης αποτελεσμάτων, και στο πεδίο εισαγωγής κειμένου θα εισάγει “Α”. Αν πάλι ο χρήστης θέλει να αναζητήσει κριτικές εστιατορίων που περιέχουν τη λέξη “πίτσα”, θα επιλέγει αρχικά “Αναζήτηση κριτικών”, στη συνέχεια θα επιλέγει “Αναζήτηση με βάση λέξη κλειδί”, θα επιλέγει τρόπο εμφάνισης αποτελεσμάτων και στο πεδίο εισαγωγής κειμένου θα εισάγει “πίτσα”.

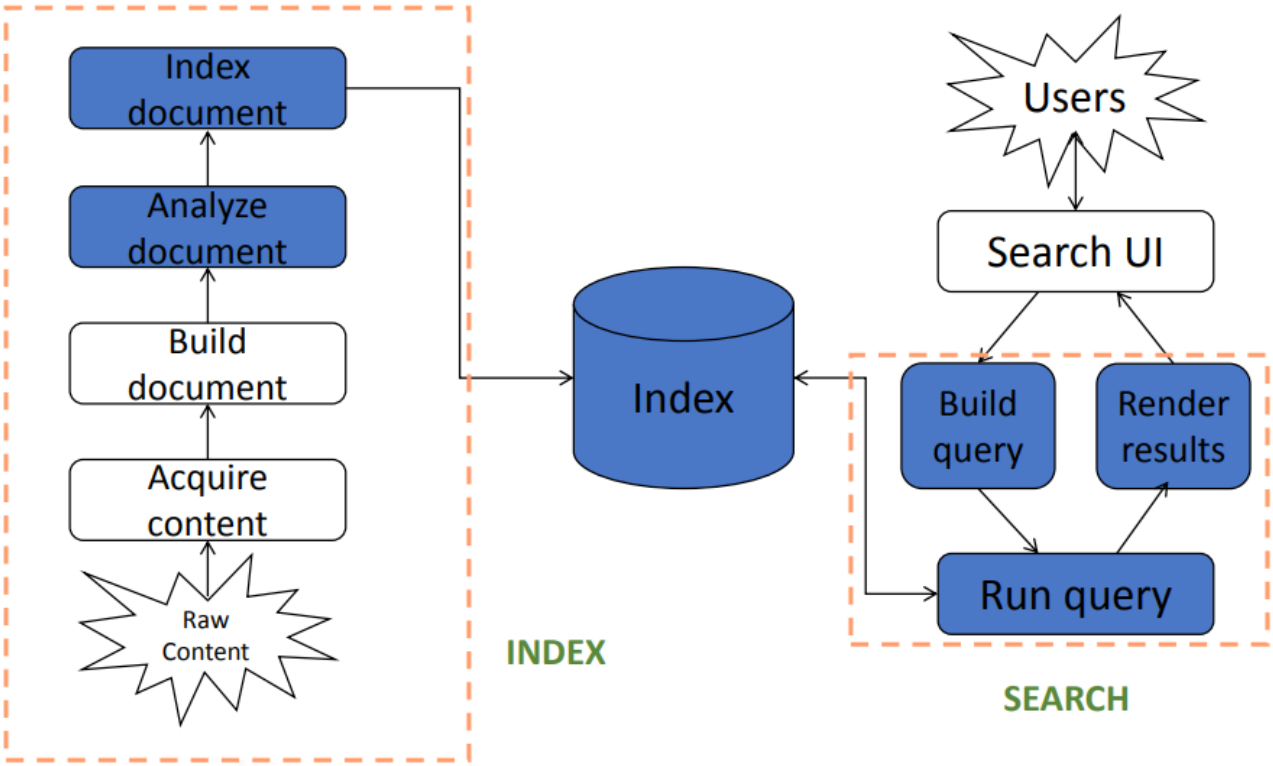
3. Ο βασικός τρόπος διάταξης και στη λειτουργικότητα 1, και στη λειτουργικότητα 2, θα είναι με βάση το κείμενο. Θα δίνεται όμως επιλογή στο χρήστη να διαλέξει διαφορετικό τρόπο διάταξης. Συγκεκριμένα, στη λειτουργικότητα 1, θα έχει επιλογή διάταξης με βάση τον αριθμό των κριτικών και με βάση τον αριθμό των αστεριών και στη λειτουργικότητα 2, θα έχει επιλογή διάταξης με βάση τη σημαντικότητα της κάθε κριτικής και με βάση το πόσο πρόσφατη είναι η κάθε κριτική. Αυτό θα επιτευχθεί με τον υπολογισμό των tf (term frequency/συχνότητα όρου), idf (inverse document frequency/αντίστροφη συχνότητα εγγράφων) και του score (βαθμός εγγράφου και ερώτησης). Επίσης, για το σκοπό αυτό θα χρησιμεύσουν τα τμήματα της Lucene: TopDocs και ScoreDocs.

4. Ο τρόπος επιλογής των αποτελεσμάτων, έτσι ώστε αυτά να είναι αντιπροσωπευτικά, θα εξαρτάται από την περίπτωση. Στην περίπτωση των εστιατορίων, η επιλογή θα γίνεται με κριτήριο οι τοποθεσίες να είναι διαφορετικές, καθώς επίσης να έχουν διαφορετικό αριθμό αστεριών. Τα αντίστοιχα κριτήρια στην περίπτωση των κριτικών εστιατορίων θα είναι οι κριτικές να αποτελούν μείγμα πιο πρόσφατων και λιγότερο πρόσφατων κριτικών και να είναι επίσης συνδυασμός κριτικών που έχουν ψηφιστεί ως useful, funny και cool.

Σύνοψη

Μία σύνοψη όλων των παραπάνω για τον τρόπο κατασκευής του συστήματος μας:

Αρχικά, θα φτιάξουμε ένα έγγραφο για κάθε εστιατόριο, συμπεριλαμβάνοντας σε αυτό τα κατάλληλα πεδία. Έπειτα θα κάνουμε Analyze και Index και θα εισαγάγουμε τα κατάλληλα δεδομένα στο λεξικό μας. Με βάση αυτά θα δημιουργήσουμε τα κατάλληλα αντεστραμμένα ευρετήρια. Όταν ο χρήστης θα δίνει μία ερώτηση, θα γίνεται η κατάλληλη επεξεργασία σε αυτήν ώστε να μπορεί να γίνει αναζήτηση στα ευρετήρια. Η επεξεργασία θα είναι η ίδια με αυτή που θα έχει ακολουθηθεί στα έγγραφα, ώστε να προκύψουν οι όροι του λεξικού και των ευρετηρίων. Τέλος, αφού γίνει η αναζήτηση, θα προβάλλονται στο χρήστη ως αποτέλεσμα τα συναφή εστιατόρια ή οι συναφείς κριτικές ανάλογα με το θέμα της αναζήτησης. Όλη η διαδικασία φαίνεται παραστατικά στο παρακάτω σχεδιάγραμμα.


Index document
Analyze document
Build document
Acquire content
Raw Content
INDEX
Index
Users
Search UI
Build query
Render results
Run query
SEARCH