

Lucene Search Engine



Ιωάννης Τσιρώνης 4908

Κωνσταντίνος Ανδρέου 4316

[GitHub](#)

[YouTube](#)

Εισαγωγή

Στόχος της εργασίας είναι να δημιουργήσουμε μια εφαρμογή που επιτρέπει την αναζήτηση τραγουδιών από ένα ευρετήριο, καθώς και σημασιολογική αναζήτηση χρησιμοποιώντας ένα Google embedding.

Η εφαρμογή αποτελείται από μία σειρά από services, γραμμένα σε Java και Python που δουλεύουν μεταξύ τους για να προσφέρουν στον χρήστη ένα πλήθος λειτουργιών σε ένα εύχρηστο περιβάλλον.

Lucene Search Service

Η κύρια υπηρεσία αναζήτησης υλοποιείται χρησιμοποιώντας Apache Lucene και είναι υπεύθυνη για την αναζήτηση αποτελεσμάτων στο ευρετήριο.

Create Index Service

Η υπηρεσία δημιουργίας ευρετηρίου δημιουργεί ένα ευρετήριο όπου εισάγει τα δεδομένα τραγουδιών για τη γρήγορη αναζήτηση τους.

Semantic Search Service

Η υπηρεσία σημασιολογικής αναζήτησης υλοποιείται σε Python χρησιμοποιώντας το Google Word2Vec embedding, εκπαιδευμένο σε πάνω 100 δισεκατομμύρια λέξεις από το Google News. Δίνει τη δυνατότητα εμφάνισης αποτελεσμάτων τα οποία παρότι δεν περιέχουν τις ίδιες λέξεις με αυτές του ερωτήματος, έχουν παρόμοια σημασία.

Backend

Η βάση που οργανώνονται όλα αυτά τα services είναι το Back-End API γραμμένο σε Django(Python web framework).

Το API λαμβάνει τα αιτήματα του χρήστη από το Front-End και καλεί τα κατάλληλα services για να τα ικανοποιήσει.

Frontend

Front-End υλοποιείται με Django-Templates και με ένα εύχρηστο τρόπο προσφέρει στο χρήστη όλες τις υπηρεσίες της Εφαρμογής.

Υπηρεσίες της Lucene Search Engine

- Υπηρεσία Αναζήτησης(Lucene Search Service) για τους στίχους, τον καλλιτέχνη, αλλά και τον τίτλο τραγουδιού
- Υπηρεσία Σημασιολογικής Αναζήτησης(Semantic Search Service)
- Ιστορικό Αναζητήσεων
- Πρόταση Αναζήτηση με βάση το Ιστορικό Αναζήτησης(History based recommendation),
- Οπτικοποίηση των αποτελεσμάτων
- Έμφαση στους ορούς της αναζήτησης που βρίσκονται στα αποτελέσματα
- Αλλαγή σελίδας(μπρος και πίσω)
- Υπηρεσία Ομαδοποίηση

Συλλογή και Προ επεξεργασία αρχείων

Η συλλογή τραγουδιών αποτελείται από 57,650 τραγούδια από το spotify_millsong_data.csv dataset που βρίσκεται στο Kaggle. Κάθε τραγούδι περιγράφεται από τον καλλιτέχνη, τον υπερ-σύνδεσμο του, τον τίτλο και τους στίχους του.

Στο στάδιο της προ επεξεργασίας των δεδομένων:

- αφαιρέθηκε το πεδίο του υπερ-συνδέσμου του
- αφαιρέθηκαν οι χαρακτήρες \n, \r και \t από όλα τα πεδία
- αφαιρέθηκαν τα τραγούδια που έχουν ακριβώς ίδιους στοίχους με άλλα τραγούδια και αυτά που έχουν ίδιο καλλιτέχνη με ίδιο τίτλο
- δημιουργήθηκε ένα αρχείο clean_songs.tsv που θα χρησιμοποιηθεί για τη δημιουργία του ευρετηρίου.

Create Index Service

Για την υπηρεσία δημιουργίας ευρετηρίου χρησιμοποιήθηκε η κλάση CreateIndex, η οποία αρχικά φτιάχνει το directory στο οποίο θα αποθηκευτεί το ευρετήριο, στον δίσκο και στην συνέχεια προσθέτει την συλλογή τραγουδιών στο ευρετήριο(Ευρετηριοποίηση).

Lucene Search Service

Για την κύρια υπηρεσία αναζήτησης χρησιμοποιείται την κλάση LuceneModule, η οποία δέχεται ως command-line arguments, το ερώτημα προς αναζήτηση, το πεδίο αναζήτησης και τον αριθμό της σελίδας εμφάνισης των αποτελεσμάτων.

Στη συνέχεια, διαβάζεται το ευρετήριο και αναζητούνται τα πιο συναφή τραγούδια ως προς το ερώτημα αναζήτησης. Οι λέξεις της ερώτησης που εμφανίζονται στα τραγούδια παρουσιάζονται με bold.

Τέλος, τα αποτελέσματα αντιστοιχούν στον αριθμό της σελίδας μεταφέρονται στο standard output.

Το Lucene Search Service εξάγεται σε ένα jar αρχείο(search_app.jar), για να μπορεί να κληθεί από το Backend.

Backend

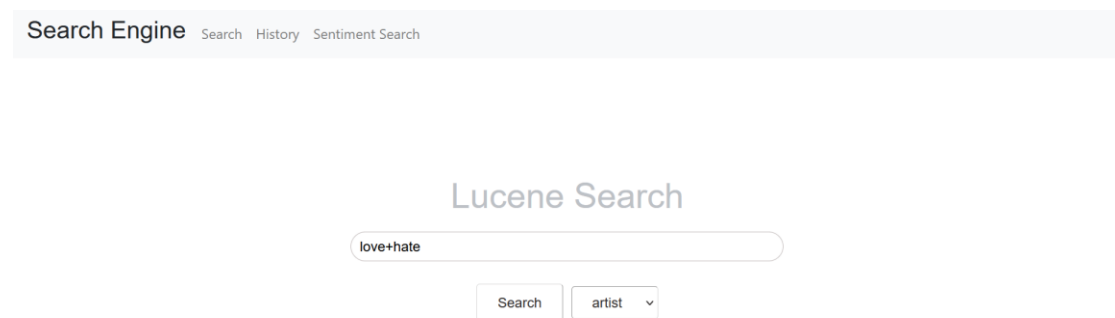
Search controller

Όταν ο χρήστης κάνει αίτημα για αναζήτηση(Εικόνα 1), ο search controller καλεί το search service μεταφέροντας του το αίτημα του χρήστη.

Το search service καλεί το Lucene Search Service για να κάνει την αναζήτηση, μεταφέροντας τα αποτελέσματα της, στο process_output service για να τα επεξεργαστεί.

Το process_output service επιστρέφει τα αποτελέσματα σε μορφή pandas DataFrame στον search controller, ο οποίος με τη σειρά του τα μετατρέπει σε JSON και τα εμφανίζει στον χρήστη(Εικόνα 2).

Αφότου τα αποτελέσματα εμφανιστούν, ο χρήστης έχει στη διάθεση του και άλλες υπηρεσίες.



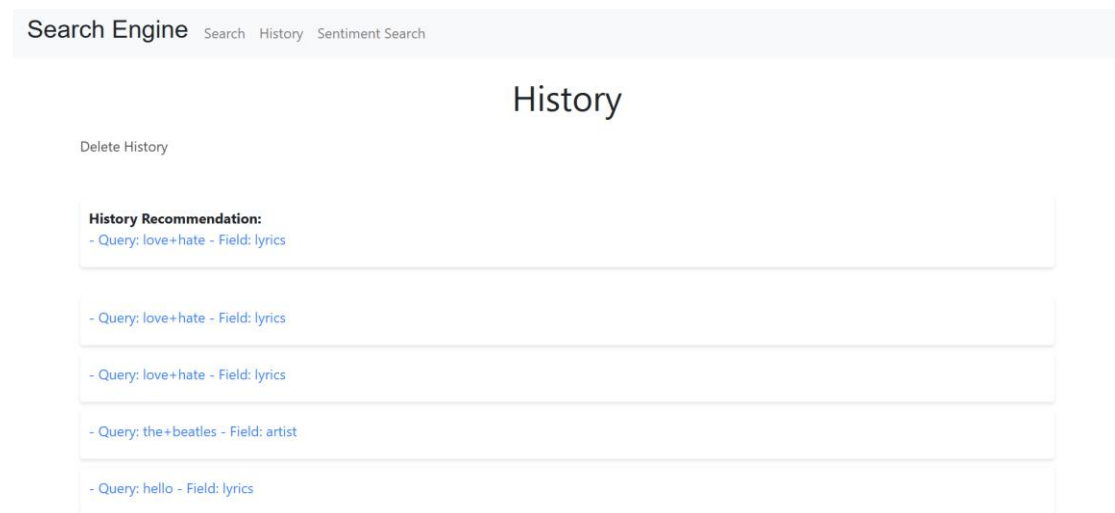
Εικόνα 1. Σελίδα Lucene Αναζήτησης.

View History controller

Κάθε φορά που ο χρήστης κάνει αίτημα αναζήτησης, αυτό αποθηκεύεται σε μια SQLite3 Βάση Δεδομένων. Όταν ο χρήστης θελήσει να δει το ιστορικό του, ο view_history controller εμφανίζει όλες τις αναζητήσεις που έχει κάνει (Εικόνα 4).

Ο χρήστης έχει τη δυνατότητα να μεταφερθεί στα αποτελέσματα της κάθε αναζήτησης επιλέγοντας αυτή.

Επίσης, έχει τη δυνατότητα να επιλέξει την Πρόταση Αναζήτησης, η οποία είναι φτιαγμένη από τις πιο σύνηθες λέξεις και πεδία αναζήτησης που έχει στο ιστορικό του.



Εικόνα 4. Σελίδα εμφάνισης ιστορικού και πρότασης αναζήτησης.

Semantic Search Service (Bonus)

Η σημασιολογική αναζήτηση, δίνει τη δυνατότητα εύρεσης αποτελεσμάτων που παρότι δεν περιέχουν τις ίδιες λέξεις με αυτές του ερωτήματος, έχουν παρόμοια σημασία. Αυτό θα βοηθήσει τον χρήστη να βρει κάποιο τραγούδι χωρίς να θυμάται τις ακριβές λέξεις (Εικόνα 5).

Για την επίτευξη αυτού του προβλήματος χρησιμοποιήθηκε το Word2Vec embedding της Google εκπαιδευμένο σε 100 δισεκατομμύρια λέξεις.

1. Οι στίχοι προ-επεξεργάζονται μετατρέποντας τις λέξεις σε πεζά γράμματα και αφαιρώντας τις πολύ μεγάλες και τις πολύ μικρές λέξεις, μέσω της συνάρτησης `preprocess_lyrics`.
2. Στη συνέχεια, για να βρεθούν οι διανυσματικές αναπαραστάσεις (δ.α.) των στίχων κάθε τραγουδιού, πρέπει:
 - Για κάθε τραγούδι να βρεθεί η δ.α. κάθε λέξης των στίχων του, με τη χρήση του `embedding`.

- Από αυτές τις δ.α. υπολογίζεται η μέση δ.α. που αναπαριστά το μέσο διάνυσμα των στίχων του τραγουδιού.
 - Το σύνολο αυτών των δ.α. για κάθε τραγούδι αποθηκεύονται στο αρχείο `encoded_data.npy`.
3. Το ερώτημα του χρήστη επεξεργάζεται και κωδικοποιείται χρησιμοποιώντας το `embedding`.
Στη συνέχεια, υπολογίζεται το cosine distance του διανύσματος της ερώτησης, με κάθε άλλη διανυσματική αναπαράσταση στίχων που προέκυψε από το παραπάνω βήμα.
Από εκεί επιλέγονται τα διανύσματα των στίχων που έχουν τη μικρότερη απόσταση από το διάνυσμα της ερώτησης και αυτά αντιστοιχίζονται στον πίνακα τραγουδιών. Τέλος, οι λέξεις της ερώτησης που εμφανίζονται στο αποτέλεσμα παρουσιάζονται με **bold** (συνάρτηση `highlight_words`).
4. Το τελικό αποτέλεσμα επιστρέφεται στον `advanced_search` controller, ο οποίος με τη σειρά του εμφανίζει τα αποτελέσματα στον χρήστη (Εικόνα 6).

Search Engine

SearchHistorySentiment Search

Sentiment Search

apple orange

Search

Εικόνα 5. Σελίδα Σημασιολογικής Αναζήτησης

Results

Page: 1

[illegible]

Εικόνα 6. Σελίδα εμφάνισης αποτελεσμάτων Σημασιολογικής Αναζήτησης